A Major Project Report

On

"RIAS (RELIABLE INTERACTIVE AR HOUSE MODEL SYSTEM)"

Submitted in partial fulfilment of the requirements for the award of the degree in

Bachelors of Technology

In

Department of Electronics and Computer Engineering

Submitted

By

KARANAM SAI GOUTHAM	**17671A1920**
B. PAVAN KUMAR CHARY	**17671A1906**
D. MOUNIKA	**17671A1917**

Under the Guidance of

Dr. Roshan Kavuri

HOD

Department of Electronics and Computer Engineering

J.B INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC Autonomous), Yenkapally, Moinabad – 500075

2015-19

“RIAS (RELIABLE INTERACTIVE AR HOUSE MODEL SYSTEM)”

A Dissertation Submitted in the Partial Fulfilment of the Academic Requirements
For the Award of the Degree of

Bachelor of Technology
In
Electronics & Computer Engineering

KARANAM SAI GOUTHAM	**17671A1920**
B. PAVAN KUMAR CHARY	**17671A1906**
D. MOUNIKA	**17671A1917**

Under the Guidance of
DR. ROSHAN KAVURI
Associate Professor

Department of Electronics and Computer Engineering
J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY
(UGC Autonomous & Accredited by NBA& NAAC, Approved by AICTE & Affiliated to JNTU, Hyderabad)
Yenkapally, Moinabad Mandal, R.R. Dist., Hyderabad-500075

2015-19

J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC Autonomous & Accredited by NBA&NAAC, Approved by AICTE & Affiliated to JNTU, Hyderabad)

CERTIFICATE

This is to certify that the dissertation work entitled **"RIAS"** was carried out by **KARANAM SAI GOUTHAM, B. PAVAN KUMAR CHARY and D. MOUNIKA** bearing **17671A1920, 17671A1906 and 17671A1917** in partial fulfilment of the requirements for the degree of Bachelor of Technology in Electronics and Computers Engineering of the J.B. Institute of Engineering and Technology, Hyderabad, during the academic year 2018-19, is a Bonafide record of work carried out under our guidance and supervision. The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

DR. ROSHAN KAVURI
Associate Professor
Internal guide

DR. ROSHAN KAVURI
Associate Professor
HOD-ECM

External Examiner

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all.

First of all, we would like to express our deep gratitude towards our internal guide**,** **DR. ROSHAN KAVURI,** HOD**,** Department of Electronics and Computer Engineering for his support in the completion of our dissertation. We wish to express our sincere thanks to **DR. ROSHAN KAVURI,** HOD, Department of Electronics and Computer Engineering and also to our Principal **Dr. P.C. KRISHNAMACHARY** for providing the facilities to complete the dissertation. We would like to thank the college management for the support in the completion of our dissertation. We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral and financial support and encouragement to achieve goals.

KARANAM SAI GOUTHAM	**17671A1920**
B. PAVAN KUMAR CHARY	**17671A1906**
D. MOUNIKA	**17671A1917**

DECLARATION

We **KARANAM SAI GOUTHAM, B. PAVAN KUMAR CHARY and D. MOUNIKA** bearing **17671A1920, 17671A1906 and 17671A1917** hereby declare that the major project entitled **"RIAS"** is submitted in the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Computer Engineering from J.B Institute of Engineering and Technology (UGC Autonomous). The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

KARANAM SAI GOUTHAM	**17671A1920**
B. PAVAN KUMAR CHARY	**17671A1906**
D. MOUNIKA	**17671A1917**

PLACE: Hyderabad
DATE: 20-6-2021

TABLE OF CONTENTS

Abstract ix
List of Figures x
List of Tables xi
Symbols and Abbreviations xii

1. INTRODUCTION 1-3

1.1. OVERVIEW 1

1.2. PROBLEM DEFINITION 2

1.3. OBJECTIVE OF THE PROJECT 2

1.4. SCOPE 3

1.5. LIMITATION OF WORK 3

1.6. EXPECTED RESULT 3

2. LITERATURE REVIEW 4-8

2.1. INTRODUCTION 4

2.2. EXISTING SYSTEM 4

2.2.1. DISADVANTAGES OF THE EXISTING SYSTEM 4

2.3. PROPOSED SYSTEM 5

2.3.1. ADVANTAGES OF THE PROPOSED SYSTEM 5

2.4. CONCEPT OF THE PROPOSED SYSTEM 5

2.4.1. MOBILE AUGMENTED REALITY 5

2.4.2. MARKER BASED AUGMENTED REALITY 6

2.4.3. MARKER BASED AUGMENTED REALITY USING ANDROID 6

2.5. AUGMENTED REALITY VS VIRTUAL REALITY 7

3. ANALYSIS 8-11

3.1. INTRODUCTION 8

3.2. REQUIREMENT SPECIFICATION 8

3.2.1. USER REQUIREMENT SPECIFICATION 8
3.2.1.1. FUNCTIONAL REQUIREMENT SPECIFICATION 9
3.2.1.2. NON-FUNCTIONAL REQUIREMENT SPECIFICATION 9
3.2.2. HARDWARE SPECIFICATION 9
3.2.3. SOFTWARE SPECIFICATION 9
3.3. SYSTEM MODULES 10
3.3.1. INITIAL DATA REQUIREMENT 11
3.3.2. INTEGRATING IN UNITY 11
3.3.3. TESTING AND DEBUGGING 11
3.3.3.1. TESTING 11
3.3.3.2. DEBUGGING 11
3.3.4. MAKING AN ANALYSIS REPORT 11
3.3.5. PRODUCT DEPLOYMENT 11
4. DESIGN 12-18
4.1. INTRODUCTION 12
4.2. ADDIE DEVELOPMENT MODEL ARCHITECTURE 12
4.2.1. ANALYSIS 12
4.2.2. DESIGN 13
4.2.3. DEVELOPMENT 13
4.2.4. IMPLEMENTATION 13
4.2.5. EVALUATION 13
4.3. RIAS DEVELOPMENT MODEL ARCHITECTURE 14
4.4. UML DIAGRAMS 14
4.4.1. USE CASE MODEL 14
4.4.1.1. USE CASE DIAGRAM 14
4.4.2. CLASS DIAGRAM 15
4.4.3. SEQUENCE DIAGRAM 16

4.4.4. COLLABORATION DIAGRAM 17

5. IMPLEMENTATION 19-33

5.1. INTRODUCTION 19

5.2. EXPLANATION OF KEY FEATURES 19

5.3. METHOD OF IMPLEMENTATION 22

5.3.1. INITIAL DATA REQUIREMENT 22

5.3.2. INTEGRATING IN UNITY 26

5.3.3. TESTING AND DEBUGGING 30

5.3.4. MAKING AN ANALYSIS REPORT 31

5.3.5. PRODUCT DEVELOPMENT 31

5.4. RIAS APP INTERFACE 32

5.5. OTHER DETAILS

5.5.1. VUFORIA TARGET UPLOAD PROCESS 33

5.5.2. CODE 35

5.5.3. UNITY ENGINE CONCEPTS 38

5.5.4. VUFORIA AUGMENTED REALITY SDK 43

5.5.5. BLENDER 2.6 46

5.5.6. ADOBE PHOTOSHOP 2021 47

6. CONCLUSION 48-49

6.1. PROJECT CONTRIBUTION 48

6.2. PRODUCT USAGE 48

6.3. FUTURE ENHANCEMENTS 48

6.4. APPLICATIONS 49

6.5. RIAS VS EXISTING SYSTEM MODELS 49

6.6. CONCLUSION 49

7. BIBLIOGRAPHY 50

ABSTRACT

In this paper, we have proposed a model for A Superimposed Interactive House Model design using AR and AI. Through which we can interact and move around the house 3d design or model**.** We are calling it as RIAS. RIAS Stands for Reliable Interactive AR house model System. In Architecture, Engineering fields building a 3d model for a house/ building is mandatory. In current technology there are many tools to design the layout of the houses and its design, such as AUTOCAD, etc. and two make it a 3d model there are quite powerful Software such as MAYA 3D, BLENDER, etc. So, by taking the 3D model using Augmented Reality we are going to generate Augments of the building using Computer Generated Graphics through Unity Engine and Vuforia Engine on the target object. So, to be precise we are going to use Marker based Augmented Reality Technology to develop this project. We use AI python bot to operate it by using voice commands this is an additional point and one of our project's premium features.

We are developing a whole new premium feature that has never been developed yet i.e., Live Home Experience. Using this feature you can walk through, in, round and outside of the house through your mobile screen. This feature really makes our project standout from all the products in the existing system. The main application of the project is Real Estate.

RIAS is designed to Public, we have a gut feeling that this product will completely revolutionize the Digital Experience and marketing approach in the field of Real Estate through the latest technology Augmented Reality, RIAS user-friendly Interface allows users to use it more, as it is more interesting, attractive and provides a real-time experience. It's because Our Project has been successful in making the user's ideology express through AR representation to the customers instead of using the paper card Brochure and able to bring new Real-world Experience through our app's premium features Live Home Experience. It helps both to the seller and the customer to have a whole new Interactive, Attractive and Safest ever Experience by going through the house they desired to buy right from their own mobile screens.

LIST OF FIGURES

Sno.	Figure no.	Figure name	Pageno.
1	Figure 2.4.1	Mobile Augmented Reality	5
2	Figure 2.4.2	Marker Based Augmented Reality	6
3	Figure 2.4.3	Marker Based Augmented Reality using Android	7
4	Figure 3.3	The Methodology Tree	10
5	Figure 4.2	ADDIE Development Model Architecture	12
6	Figure 4.3	RIAS Development Model Architecture	14
7	Figure 4.4.1	Use case Diagram for RIAS Model	15
8	Figure 4.4.2	Class Diagram for RIAS Model	16
9	Figure 4.4.3	Sequence Diagram for RIAS Model	17
10	Figure 4.4.4	Collaboration Diagram for RIAS Model	18
11	Figure 5.2	Unity Engine Interface	19
12	Figure 5.2.2	Vuforia Engine Logo	20
13	Figure 5.2.3	Visual Studio and Visual Studio Code Logo	20
14	Figure 5.2.4	Adobe Photoshop 2021 logo	21
15	Figure 5.2.5	Blender 2.6 Logo	21
16	Figure 5.2.5	Android Mobile sample	21
17	Figure 5.3.1	Methodology Tree Phase 1	22
18	Figure 5.3.1(a)	House Model 1	22
19	Figure 5.3.1(a)	House Model 2	22
20	Figure 5.3.1(b1)	Image targets Brochure House1	23
21	Figure 5.3.1(b1)	Image targets Brochure House2	23
22	Figure 5.3.1(b2)	Vuforia Developer Portal – Target Manager	24
23	Figure 5.3.1(b3)	Target Image Features	24
24	Figure 5.3.1(c)	3D Models of House 1 and 2	25
25	Figure 5.3.2	Methodology Tree Phase 2	26
26	Figure 5.3.2(a)	Unity UI - RIAS model development at the end of Phase 2 Step 1	27

27	Figure 5.3.2(b1)	Unity UI - RIAS model development at the end of Phase 2 Step 2	28
28	Figure 5.3.2(b1)	Unity Game View of Live Home Experience Feature	28
29	Figure 5.3.2(c)	Training RIAS bot from wit.ai	29
30	Figure 5.3.3	Methodology Tree Phase 3	30
31	Figure 5.3.4	Methodology Tree Phase 4	31
32	Figure 5.3.5	RIAS app in Sample Mobile	31
33	Figure 5.4(a)	RIAS app UI – Intro Scene	32
34	Figure 5.4.(b)	RIAS app UI – Image target 1 AR scene	32
35	Figure 5.4.(c)	RIAS app UI – Live Home Experience	32
36	Figure 5.5.1(a):	Vuforia Login page	33
37	Figure 5.5.1(b):	License Key Activation page	33
38	Figure 5.5.1(c):	License Key page	34
39	Figure 5.5.1(d):	Create Database	34
40	Figure 5.5.1(e):	Vuforia Target manager	35
41	Figure 5.5.3:	Mesh Renderer	39
42	Figure 5.5.3(h-a):	Screen Space - Overlay	41
43	Figure 5.5.3(h-b):	Screen Space - Camera	42
44	Figure 5.5.3(h-c):	World Space - Camera	43
45	Figure 5.5.4:	AR Camera Configuration	44
46	Figure 5.5.5(a):	Blender 2.6 Interface	46
47	Figure 5.5.5(b):	House model-2	46
48	Figure 5.5.6(a):	Adobe Photoshop 2021 Image Target Design	47
49	Figure 5.5.6(b):	Adobe Photoshop 2021 App UI	47
50	Figure 6.3	Future Enhancements	48

LIST OF TABLES

Sno.	Table no.	Table name	Pageno.
1	Table 2.5	Differences Between AR and VR	7
2	Table 6.5	RIAS vs Existing system models	34
3	Table 5.5.4	AR SDKs Comparision table	45

SYMBOLS AND ABBREVIATIONS

Sno.	Symbol	Abbreviation
1	RIAS	Reliable Interactive AR house model System
2	AR	Augmented Reality
3	VR	Virtual Reality
4	AI	Artificial Intelligence
5	NLP	Natural language Processing
6	UI	User Interface
7	FPP	First Person Perspective
8	LHE	Live Home Experience
9	CV	Computer Vision
10	PS	Adobe Photoshop
11	VS	Visual studio
12	2D	2 - Dimensional
13	3D	3 - Dimensional
14	ADDIE	Analyze, Design, Develop, Implement & Evaluate
15	UML	Unified Modelling language

1. INTRODUCTION

1.1 Overview

Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world. It is an enhanced version of the real physical world through the use of visual elements, sound or other sensory stimuli of interactive technology that give us a new way to interact in direct view of an existing environment. It allows us to create the 3D object directly into physical things or fused together in real-time. In this era of technology, there are various applications that integrated with augmented reality technology.

The primary value of augmented reality is the manner in which components of the digital world blend into a person's perception of the real world, not as a simple display of data, but through the integration of immersive sensations, which are perceived as natural parts of an environment. The earliest functional AR systems that provided immersive mixed reality experiences for users were invented in the early 1990s, starting with the Virtual Fixtures system developed at the U.S. Air Force's Armstrong Laboratory in 1992. Commercial augmented reality experiences were first introduced in entertainment and gaming businesses. Subsequently, augmented reality applications have spanned commercial industries such as education, communications, medicine, and entertainment. In education, content may be accessed by scanning or viewing an image with a mobile device or by using marker less AR techniques.

Augmented reality is used to enhance natural environments or situations and offer perceptually enriched experiences. With the help of advanced AR technologies (e.g. adding Computer Vision (CV), incorporating AR cameras into smartphone applications and object recognition) the information about the surrounding real world of the user becomes interactive and digitally manipulated. Information about the environment and its objects is overlaid on the real world. This information can be virtual. Augmented Reality is any experience which is artificial and which adds to the already existing reality. or real, e.g., seeing other real sensed or measured information such as electromagnetic radio

waves overlaid in exact alignment with where they actually are in space. Augmented reality also has a lot of potential in the gathering and sharing of tacit knowledge. Augmentation techniques are typically performed in real time and in semantic contexts with environmental elements. Immersive perceptual information is sometimes combined with supplemental information like scores over a live video feed of a sporting event. This combines the benefits of both augmented reality technology and heads up display technology Hud.

1.2 Problem Definition

We use the House/The Real Estate Home Brochure only to know about basic information for the house. Using this existing system pattern i.e., in a normal Brochure we can only read the information that is printed on the card which is pretty boring and unattractive. We cannot convey the information or disclose more details regarding the house as much as we would like to because space becomes a constrain. To overcome all these, we can make an interactive brochure through which we can interact with the buttons, play videos, animations, view inside the house, operate through voice commands, etc. We are going to use Augmented Reality to develop this project.

1.3 Objective of the Project

To develop a target based Reliable Interactive AR house model system (RIAS app) through which we can play videos, animations, interact by pressing buttons, view the house through our own mobiles etc, with the help of cutting-edge technology Augmented Reality. The main Goal of the project is to Develop an AR house model app and target brochure which is interactive, cost efficient, secure and easy to use and thus makes the business more effective, promising and standout in the market. This works by viewing the Brochure through scanning from the app that we are going to develop using AR Engine.

1.4 Scope

The scope of this study is divided into 2:

a. **User**

The content is generally suitable for all ages and they also can experience to see the virtual object appear in real-world with the animation.

b. **Application**

Augmented reality presented in 3D mobile application by using a smartphone. Through application we can perform all the interactions

1.5 Limitation of Work

The limitations in conducting this application the users need a mobile-based application for start the augmented reality and only android version can access the application. The technique of the application is marker based mobile AR.

1.6 Expected Result

The expected result of RIAS application after completing the development is the user can view and express the information clearly with a good interface. At the same time, the user can emphasize the button functionality on wide space. By using the app, the user can experience visual animations and can play the videos, use voice controls and access its other premium features.

2. LITERATURE REVIEW

2.1 Introduction

Literature review means searching, collecting and analysing any issues related from the previous journal or research paper. The information that has been gathered is about what method and what contributions that the research has been achieved in improving the uses of the application. This chapter will be discussing the report analysing the approach and techniques that are going to be used in completing the RIAS project. This literature is made for identifying the weakness or lacking of related literature review sources.

2.2 Existing System

Using this existing system pattern i.e., in a normal Brochure we can only read the information that is printed on the card which is pretty boring and unattractive. We cannot convey the information or disclose more details regarding the house as much as we would like to because space becomes a constrain.

2.2.1 Disadvantages of the Existing System

It is only possible to show a 3D model of your house in a 2D view through pictures of the house and all. It is also not possible to view the interior of the house as a walkthrough. We can only show them pictures of it. Even if we dare and print the pictures on the brochure it becomes so lengthy and more pages are needed to show them. In current generation no one wants to read a book that looks like a brochure right! It is obvious that we cannot play videos, use voice commands, view the house as a 3D model, go through the house inside, round and outside as it a piece of paper.

2.3 Proposed System

The Primary goal of this project is to Comparison overcome all these we can make an interactive brochure through which we can interact with the buttons and do a bunch of other visual perceptions.

2.3.1 Advantages of the proposed system

The brochure is made interactive s through which we can interact with the buttons, play videos, animations, view inside the house, operate through voice commands, etc. We are going to show everything virtually on the mobile screen with enhanced Computer-Generated graphics which will be visually eye-catching.

2.4 Concept of the proposed System

2.4.1 Mobile Augmented Reality

As computers increase in power and decrease in size, new mobile, wearable, and pervasive computing applications are rapidly becoming feasible, providing people access to online resources always and everywhere. This new flexibility makes possible new kind of applications that exploit the person's surrounding context. Augmented reality (AR) as an emerging technology in the mobile computing domain is becoming mature enough to engender publicly available applications for end users. Various commercial applications have recently been emerging in the mobile consumer domain at an increasing pace.

Figure 2.4.1: Mobile Augmented Reality

2.4.2 Marker Based Augmented Reality

A marker is a piece of paper with black and white markings. This is used to display the augmented object on your mobile phone's screen. Marker-based augmented reality uses a camera and a visual marker which determines the centre and range of its spherical coordinate system. Once the marker is present one can view augmented objects.

A camera is used with AR software to detect augmented reality markers as the location for virtual objects. The result is that an image can be viewed, even live, on a screen and digital assets are placed into the scene at the location of the markers. Limitations on the types of augmented reality markers that can be used are based on the software that recognizes them. While they need to remain fairly simple for error correction, they can include a wide range of different images.

Figure 2.4.2: Marker Based Augmented Reality

2.4.3 Marker Based Augmented Reality Using Android

Augmented Reality or AR is an emerging technology in which one's perception of the real-time environment is enhanced by superimposing computer-generated information such as graphical, textual, or audio content, as well as objects onto a display screen. The proposed application is an android mobile based application which will be compatible with all the existing and upcoming versions of the operating system. The idea is to allow the user to view the virtual object in the real world using a Marker based AR system.

The user could provide images of the object which would be the front, back, top, bottom, left and right-side pictures of the object. They will be placed onto a 3D cube which will

make up the complete virtual object. Thus, an extended environment will be created through the amalgamation of real world and generated object and it will appear as though the real-world object and virtual object coexist within the environment. The advantages of this application as compared to the already existing 2D application are that it would display object in 3D and enable the user to rotate it virtually. It is inexpensive as the user need not actually purchase the object to see how it fits in the environment, instead he can try before the purchase itself.

Figure 2.4.3: Marker Based Augmented Reality using Android

2.5 Augmented Reality vs Virtual Reality

Augmented Reality	Virtual Reality
The system augments the real-world scene.	Completely immersive virtual environment.
In AR User always have a sense of presence in the real world	In VR, visual senses are under control of the system
AR is 25% virtual and 75% real	VR is 75% virtual and 25% real
No AR headset is needed.	Some VR headset is needed. Ex: Oculus Rift, PlayStation VR, etc.,
It is used to enhance both real and virtual worlds.	It is used to enhance fictional reality for the gaming world.
With AR, end-users are still in touch with the real world while interacting with virtual objects nearer to them.	By using VR technology, VR user is isolated from the real world and immerses himself in a completely fictional world.

Table 2.5: Differences between AR and VR

3. ANALYSIS

3.1 Introduction

Requirement Specification is the part of the project which gives the details about the hardware and software requirements of the project. It also details the features of the programming language used. So, in this chapter we will debrief all the necessary requirements including the suggested AR approach. In order to implement this project, the choice of processor with maximum possible speed is made. There should be sufficient memory to store data and software tools for efficient processing.

3.2 Requirement Specification

Understanding user requirements is an integral part of systems design and is critical to the success of a support system (Shen et al., 2004). The ISO 13407 standard specifies needs and requirements. A customer-friendly support-system design begins with a thorough understanding of the needs and requirements of the customer. The benefits can include increased productivity, enhanced quality of work, reductions in support and training costs and improved user satisfaction. Requirement's analysis is not a simple process.

3.2.1 User Requirement Specification

User requirements are typically written when discussing the use cases for a project. The statement of user requirements describes what the user expects from the system. This may form part of a proposal for the work or it may be part of an creating a model through vast series of steps and writing down the functional and non-functional requirements of the user. Typically, 'high level' discussions relate to 'functional' issues. It is particularly important to get the user requirement correct because:

- It gives the user(s) confidence in what you can do;
- It enables you to fix the scope of the project early on.
- It ensures that you are able to proceed to the next phase.

3.2.1.1 Functional Requirement Specification

- Gather images to make them as Image Targets.
- Making the image targets.
- Dropping everything in Unity and do the AR work.
- Coding in C# to make the app work in real-time.

3.2.1.2 Non-functional Requirement Specification

- **Accuracy:** Accuracy is major part as we require accurate results.
- **Adaptability:** The model should adapt automatically according to user input.
- **Complete:** Requirements must cover all areas of concern, including all phases of the Execution process.
- **Clear:** Requirements should not leave anyone guessing what is required.
- **Concise:** Unnecessary requirements should be omitted, and the wording of requirements should be concise.
- **Feasibility Study:** The model to be trained on data have to be feasible.
- **Security:** Security has to be provided to the data model in case of sensitive data.
- **Testable:** Testable requirements can be measured in order to determine if the design goal is met.

3.2.2 Hardware Specification

Hard Disk	**:**	500 GB or more.
Processor	:	Preferably Intel i3 or Amd Ryzen 3 and above
Ram	:	4 GB RAM or more.

3.2.3 Software Specification

Operating System	**:**	Windows 10
Technologies	**:**	Unity, Blender, Adobe Photoshop.
Programming languages	**:**	C# Programming
Text Editor	**:**	Visual Studio Code / IDE

3.3 System Modules

1. Initial Data Requirement
2. Integrating in Unity
3. Testing and Debugging
4. Making an Analysis Report
5. Product deployment

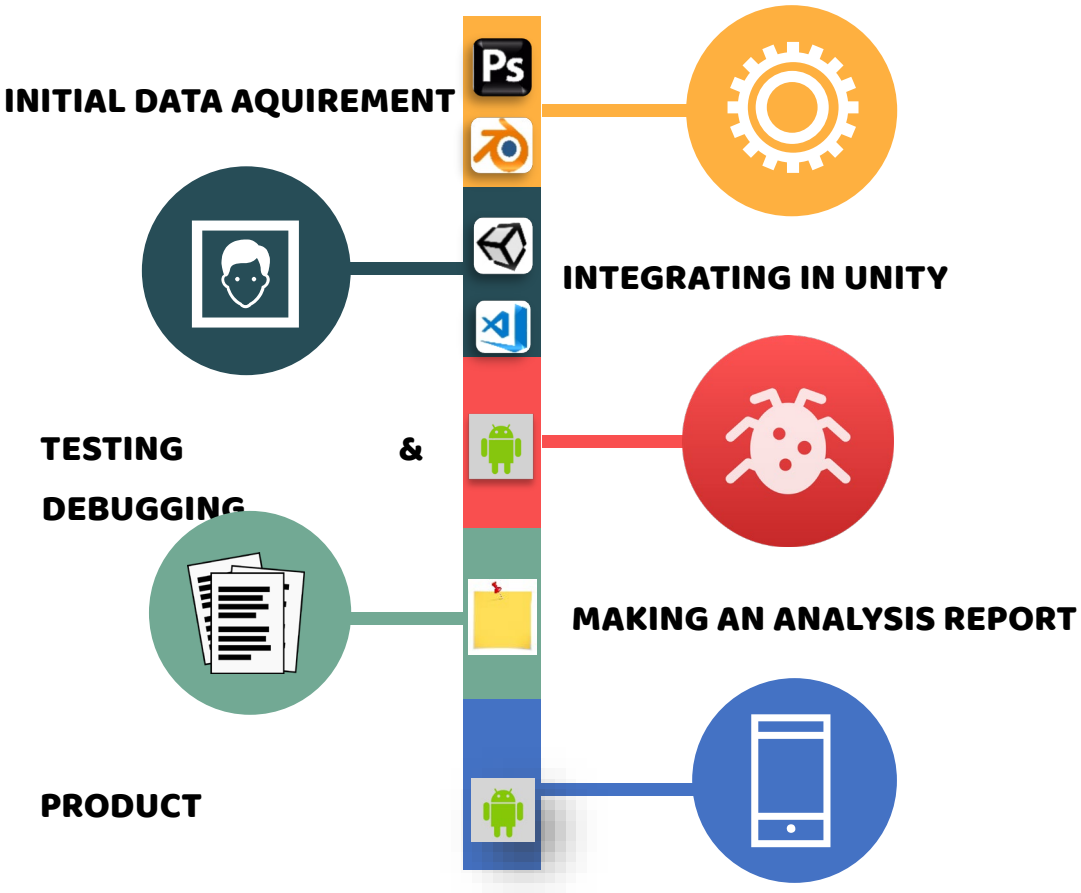


Figure 3.3: The Methodology Tree

3.3.1 Initial Data Requirement

This is the first phase of the process. Here we first gather the required house images, I have gone through the unity community and gather the house images. The very next step is to make the image targets and upload them in Vuforia Developer Database to proceed for the AR work. Later Make 3D low-poly models in Blender 3D or gather it from Unity Community.

3.3.2 Integrating in Unity:

This is the second phase of the process. Now that we have all the required Target Data it's now time to integrate everything in the Unity Software and do all AR work. We are going to use C# programming language to perform the necessary coding. This is where we develop Computer Graphics that are required for our RIAS app through Vuforia Augmented Reality Software Development Toolkit.

3.3.3 Testing and Debugging:

This is the third phase of the process. This has two phases:

3.3.3.1 Testing:

Now that we have developed everything, we need to build it as an android application and test the functionality of the app.

3.3.3.2 Debugging:

Then if any errors are found we need to debug them back in the unity or wherever the errors can be removed, then rebuild the app and retest it.

This process continues until the app gets error free and is perfect for using!

3.3.4 Making an Analysis report:

This is the fourth phase. After testing and debugging the analysis is complete. We are going to make User manual, and bunch of other things and make it ready for the deployment.

3.3.5 Product Deployment:

Then building the model as an Android Application and deploying it in the google Cloud.

4. DESIGN

4.1 Introduction

In this chapter, it will explain the specific details on the Design methodology being used in order to develop this project. In order to make sure the project is on the right path; Design methodology plays an important role as a guide for the project complete and working well as plan. The methodology for this application is ADDIE development model.

4.2 ADDIE Development Model Architecture

The phase in ADDIE development model is: Analysis Phase, Design Phase, Development Phase, Implementation Phase and Evaluation Phase. All the methods and processes undertaken in ADDIE development model are more visible.

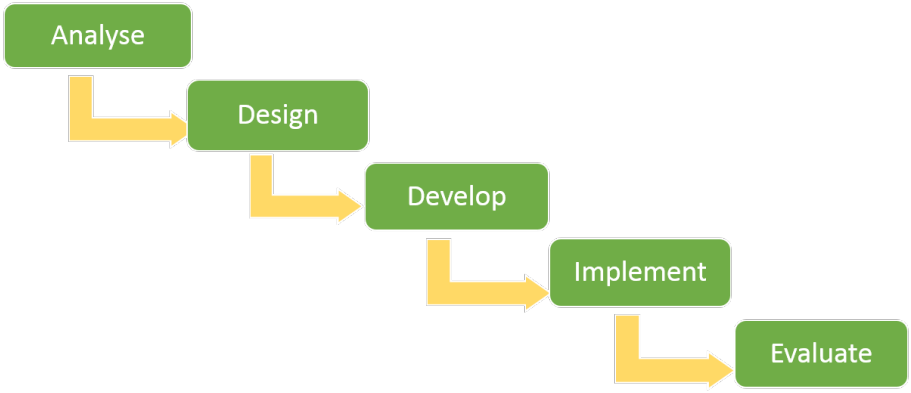


Figure 4.2: ADDIE Development Model Architecture

4.2.1 Analysis

In this first phase, ideas are collected and categorized, identify the problem, identify the problem and determine possible solutions. The main objective of this phase is to come out with new idea and improvements to the current complaint application. The analysis is also carried out by doing literature review on existing any augmented reality project. Through the analysis, the weakness of the existing augmented reality can be identified. All information gathered were used to develop goals and objective of developing AR application. The software and hardware requirement are also determined in this phase.

4.2.2 Design

In this phase, the idea application is development into an initial design of the application. This phase will explain on the design of application will develop in the form of 3D prototype. A very importance part of the design phase is to create the storyboard described flow of the application.

4.2.3 Development

The Development stage builds on both Analysis and Design phase. This phase will develop the instruction, all media that will be used in the instruction and any supporting documentation. During the development phase, the actual course materials are created. In this phase, all steps in development of the project are collected, prepared, created and ready to be tested. In this stage, designers make use of the data collected from the two previous stages, and use this information to create a program that will relay what needs to be taught to participants. If the two previous stages required planning and brainstorming, the Development stage is all about putting it into action. This phase includes three tasks, namely drafting, production and evaluation.

4.2.4 Implementation

In this phase, the other word is testing phase. Testing is one of the most important in ADDIE development model. The application must be tested before release to the user and sent to the supervisor for feedback. After feedback is received from the supervisor, the required changes are implemented through the development phase. When the second prototype is ready, it is integrated with the first prototype, tested and the send to the user. The development, prototyping and testing phase are repeated until the final application is ready. The final application is sent to the supervisor for a final feedback.

4.2.5 Evaluation

Evaluation is the final phase of the ADDIE development model. After the testing is completed and the final feedback is obtained from the supervisor, the application us ready for the evaluate by the professional panel.

4.3 RIAS Development Model Architecture

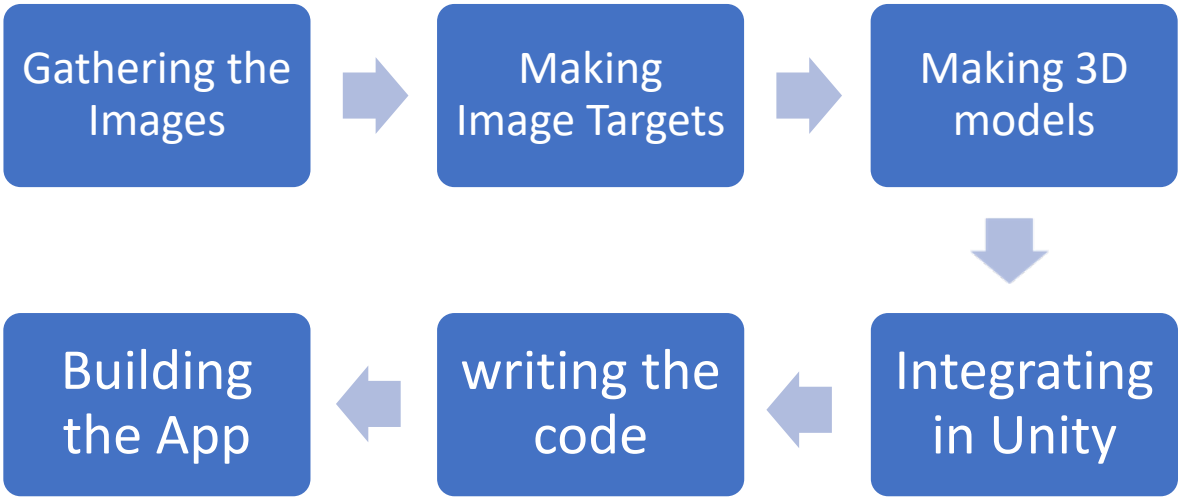


Figure 4.3: RIAS Development Model Architecture

4.4 UML Diagrams

The unified Modelling Language (UML) is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of software intensive system.

The primary goals in the design of the UML were:

1. Provide users with a ready-to-use, expressive visual modelling languages they can develop exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.

4.4.1 Use Case Model

4.4.1.1 Use case diagram:

Use case diagrams represent the functionality of the system from a user point of view. A Use case describes a function provided by the system that yields a visible result for an actor. An actor describes any entity that interacts with the system. The actors outside the boundary of the system, whereas the use cases are inside the boundary of the system

The actors in Bank defaulters Detection are:

- User
- Scanning of Image Target
- Image target Detection
- Displaying AR view in mobile
- Taking further inputs from user
- Displaying the features
- User Output

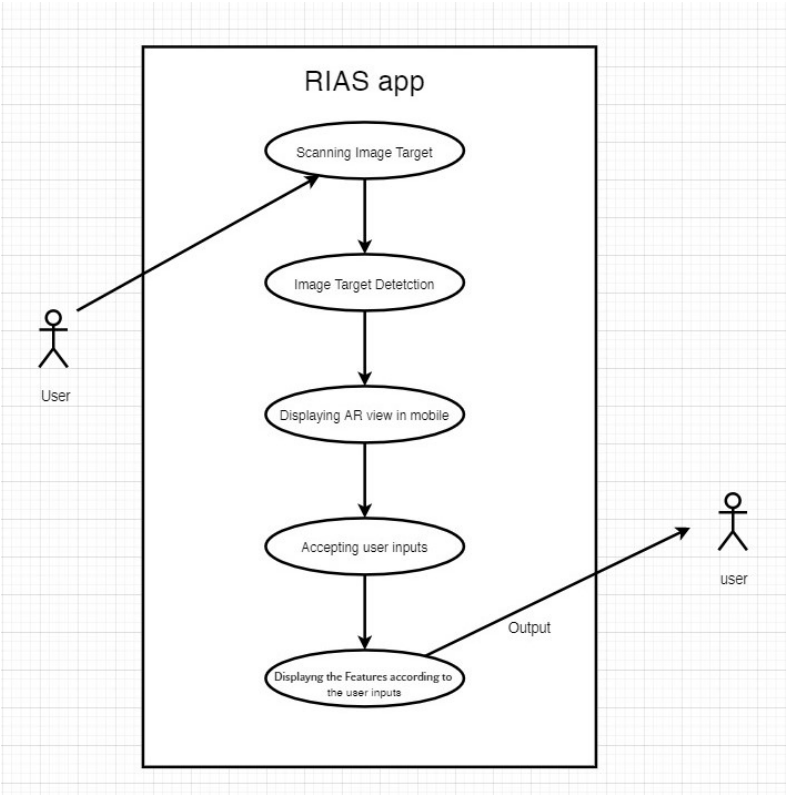


Figure 4.4.1.1: Use case Diagram for RIAS Model

4.4.2 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints.

Classes:

- User
- RIAS
- Target Image

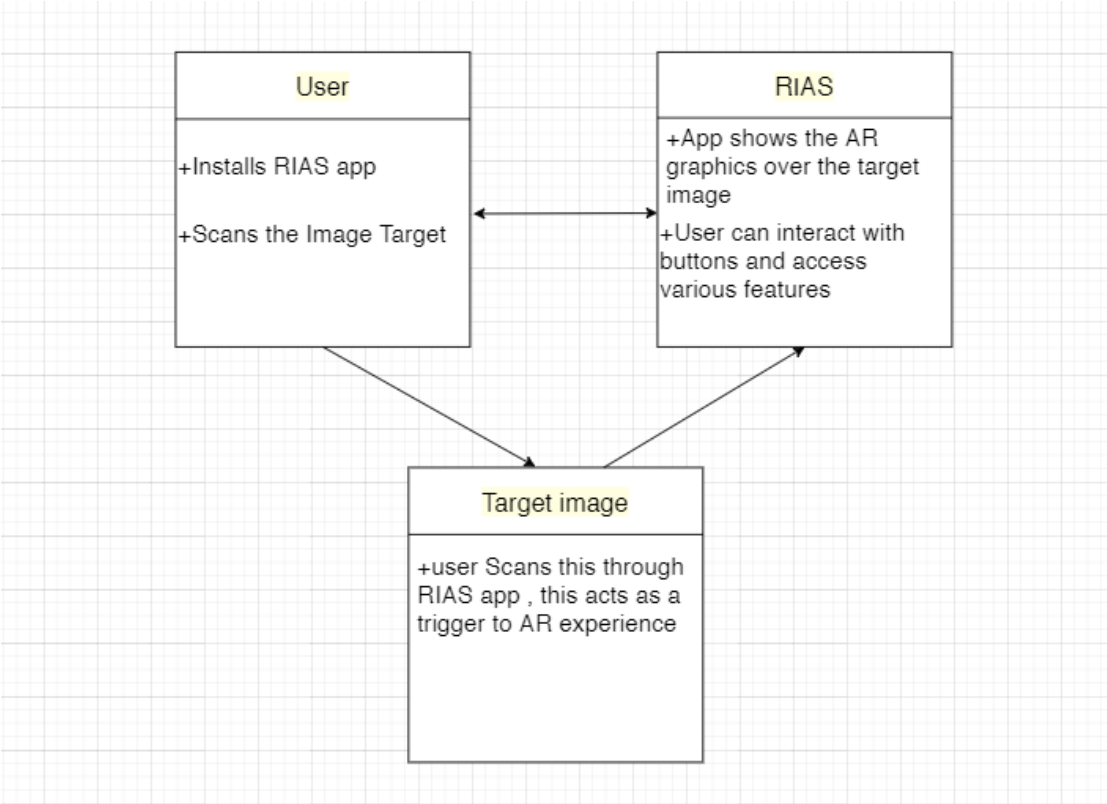


Figure 4.4.2: Class Diagram for RIAS Model

4.4.3 Sequence diagram:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. The sequence of object interactions is

- Installing RIAS app
- Scanning the Image Target through RIAS app
- Checking the Image target
- Displaying AR view
- Displaying other features based on User inputs

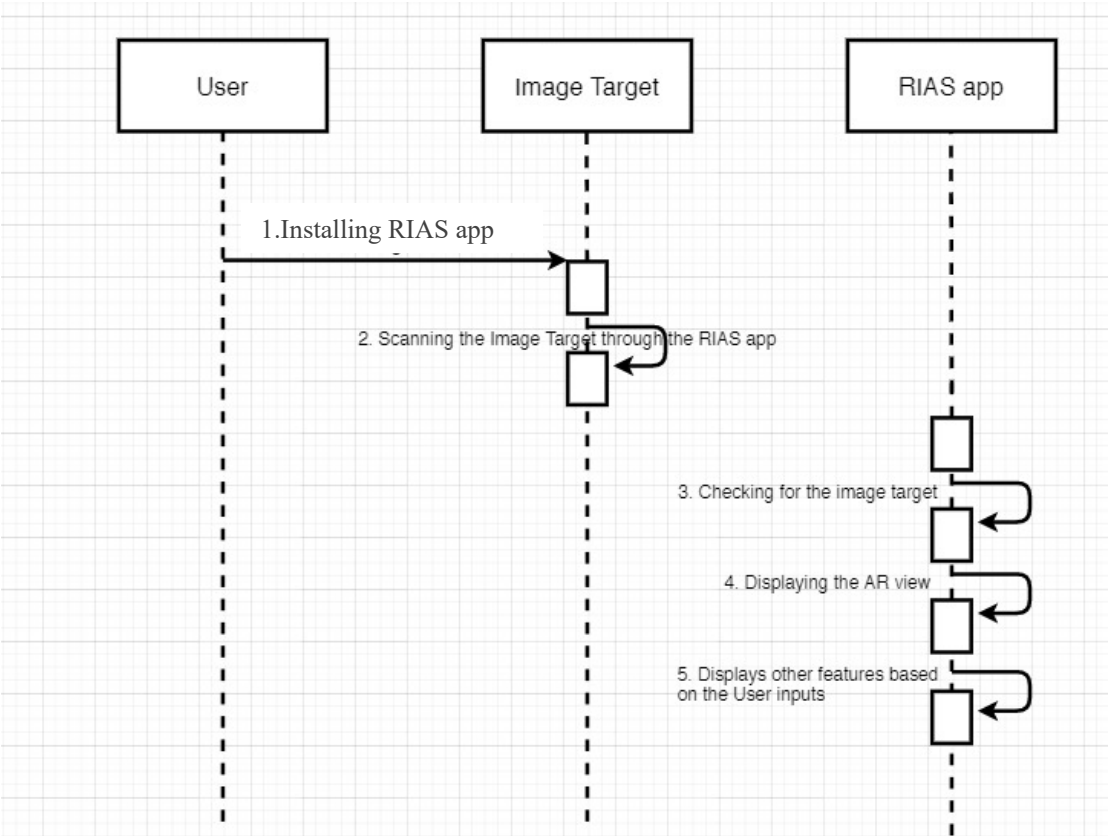


Figure 4.4.3: Sequence Diagram for RIAS Model

4.4.4 Collaboration diagram

A collaboration diagram is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users can benefit from this collaboration. A collaboration diagram often comes in the form of a visual chart that resembles a flow chart.

The various objects in Bank Defaulters detection are:

- User
- Data set collection
- Pre-processing
- Trained dataset
- Testing dataset
- Feature extraction Classification
- Accuracy of result.

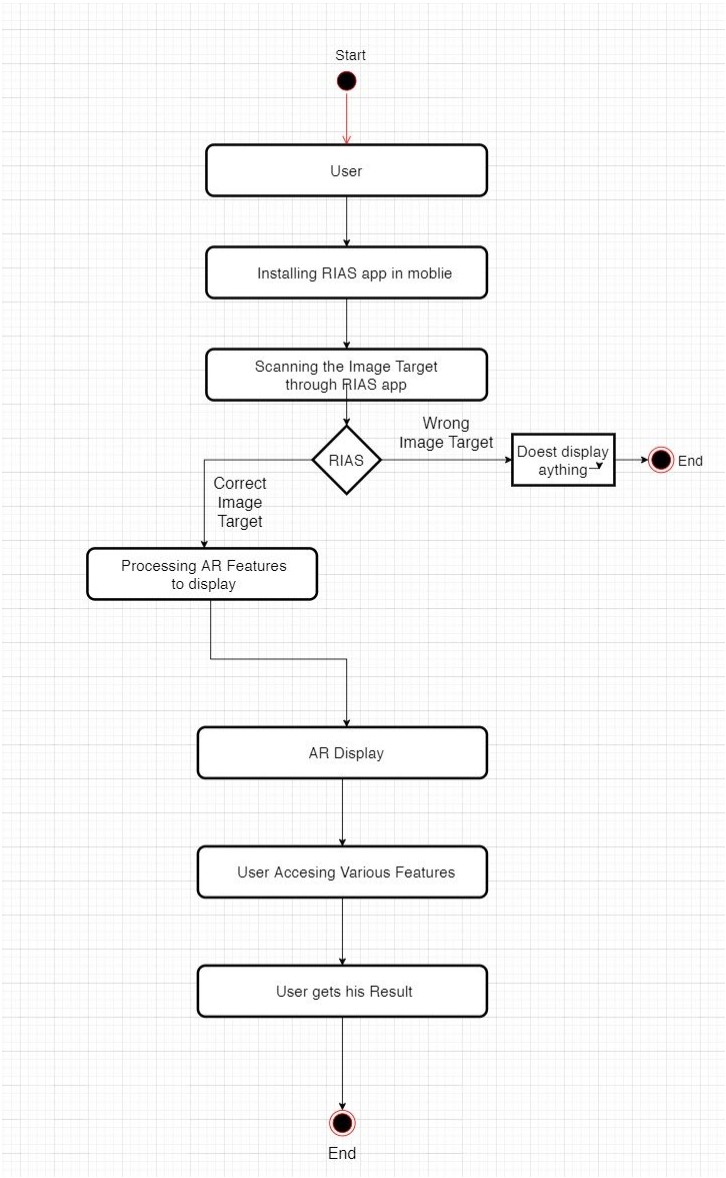


Figure 4.4.4: Collaboration Diagram for RIAS

5. IMPLEMENTATION

5.1 Introduction

In this chapter we are going to bind together all of the concepts that had specified till now and implement them through C# programming and execute them.

5.2 Explanation of key features

1. Unity 3D:

Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as a Mac OS X-exclusive game engine. The engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. It is particularly popular for iOS and Android mobile game development and used for games such as Pokémon Go, Monument Valley, Call of Duty: Mobile, Beat Saber and Cuphead. It is cited to be easy to use for beginner developers and is popular for Indie game development.

The engine can be used to create three-dimensional (3D) and two-dimensional (2D) games, as well as interactive simulations and other experiences. The engine has been adopted by industries outside video gaming, such as film, automotive, architecture, engineering and construction. Several major versions of Unity have been released since its launch.

unity

Figure 5.2: Unity Engine Interface

2. Vuforia Engine:

Vuforia Engine is a software development kit (SDK) for creating Augmented Reality apps. Developers can easily add advanced computer vision functionality to any application, allowing it to recognize images and objects, and interact with spaces in the real world. The **Vuforia** Area Target Creator application allows **developers** to easily generate an Area Target using a depth-enabled mobile device, such as the new iPads and iPhones (coming soon) with LIDAR scanners. This is essential to make our AR project in unity. We are going to integrate it in Unity and develop our RIAS app.

Figure 5.2.2: Vuforia Engine Logo

3. Visual Studio IDE:

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. The most basic edition of Visual Studio, the Community edition, is available free of charge. The Visual Studio IDE (integrated development environment) is a software program for developers to write and edit their code. Its user interface is used for software development to edit, debug and build code. We are using it because it works well with unity. We can use any text editor but we are choosing Visual studio.

Visual studio Code:

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. You can rather use this too no problem!

Figure 5.2.3: Visual Studio and Visual Studio Code Logo

4. Adobe Photoshop:

Adobe Photoshop is a raster graphics editor developed and published by Adobe Inc. for Windows and macOS. By using this we are going to Develop the Image targets which is the initial step of the project.

Figure 5.2.4: Adobe Photoshop 2021 Logo

5.Blender 2.6:

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, virtual reality, and computer games. To develop the house pictures into real 3D house models we are going to use this. However, we can do the job by obtaining those models from online if available.

Figure 5.2.4: Blender Logo

5. Android mobile v5 or higher:

For testing and debugging purpose we are going to need an android mobile. The Software version must be v5 or higher.

Figure 5.2.5: Android Mobile sample

5.3 Method of Implementation and code:

5.3.1. Initial Data Requirement:

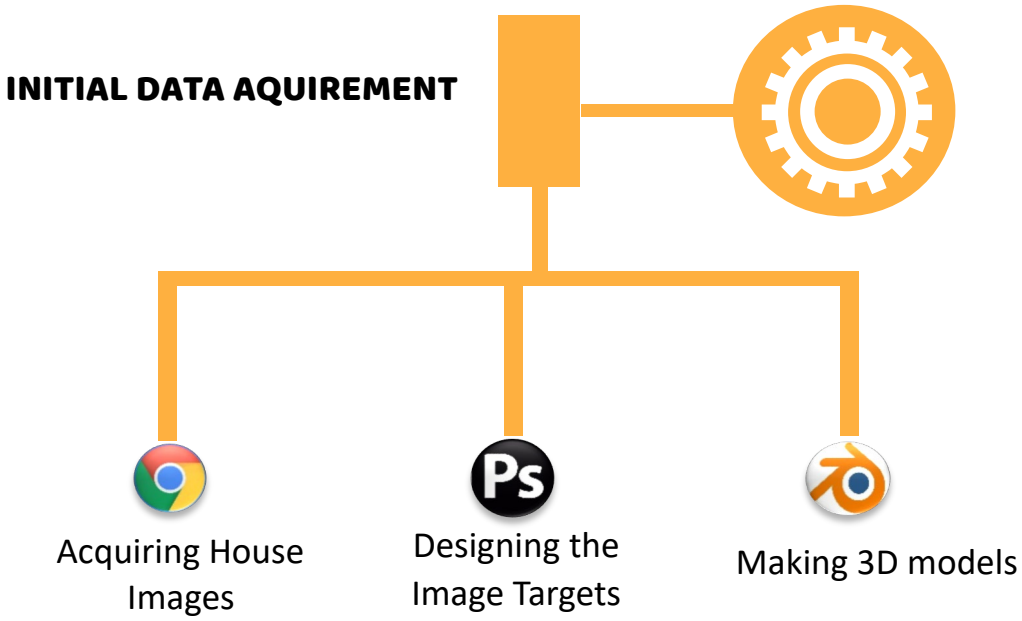


Figure 5.3.1: Methodology Tree Phase 1

There are 3 steps involved to complete this first phase they are:

a) Acquiring the House Images:

Here are going to collect the house images that we are going to use in our app. We are using Unity community to gather them. The house images that have been gathered are shown below:

Figure 5.3.1(a): House Model 1

Figure 5.3.1(a): House Model 2

b) Designing the Image Targets:

1)Making Image targets in Photoshop:

As we know that our App works on marker based Augmented Reality, Image Targets are must and should to trigger the AR experience. We are going to use Adobe Photoshop to design the image targets. We are designing 3 Image targets for each house Since we are making two houses the number of image targets should be designed are 6.

Figure 5.3.1(b1): Image targets Brochure House1

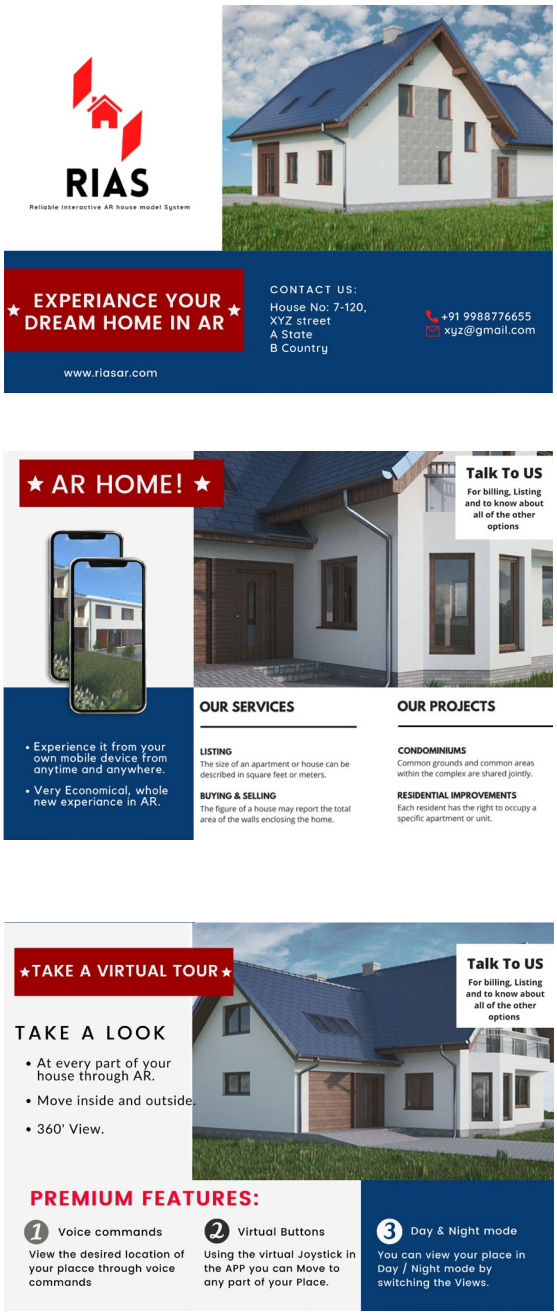


Figure 5.3.1(b1): Image targets Brochure House2

2) Generating Vuforia Data Base for our Image targets:

As we know that our App works on marker based Augmented Reality, Image Targets are must and should to trigger the AR experience. We are going to use Adobe Photoshop to design the image targets. We are designing 3 Image targets for each house Since we are making two houses the number of image targets should be designed are 6.

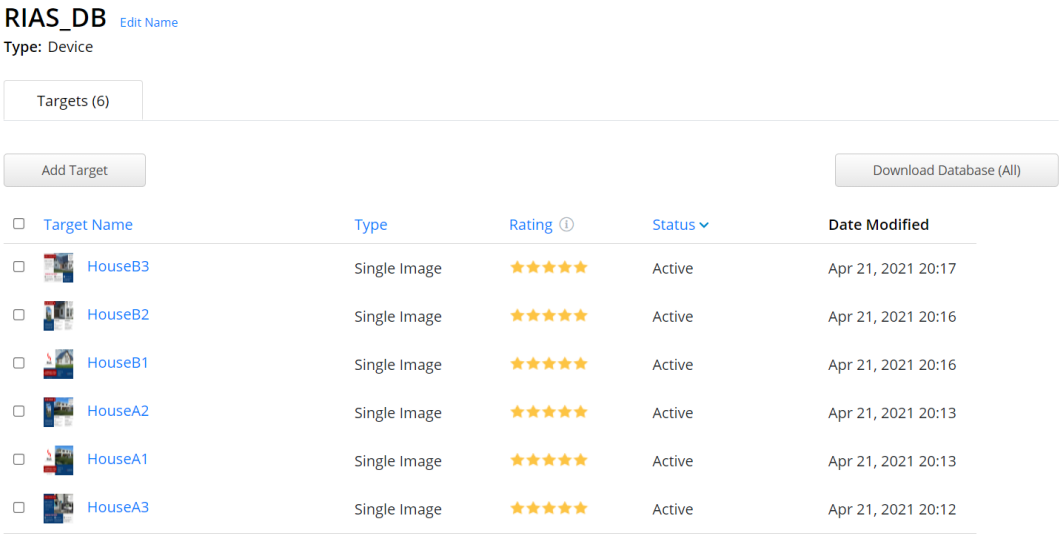


Figure 5.3.1(b2): Vuforia Developer Portal – Target Manager

3) Target Image Features:

There are two important points that we need to take care of:

i) Rating: This is very important factor because it reflects the quality of the image. The minimum rating Requirement for good AR performance is At least Three. As u can see in the above figure the rating is Five so we are Very fine to go into next process.

ii) Truce Points: These are the points where the The Augmented reality shows up when they are scanned through the mobile. The yellow crosses in the figure below represents truce points.

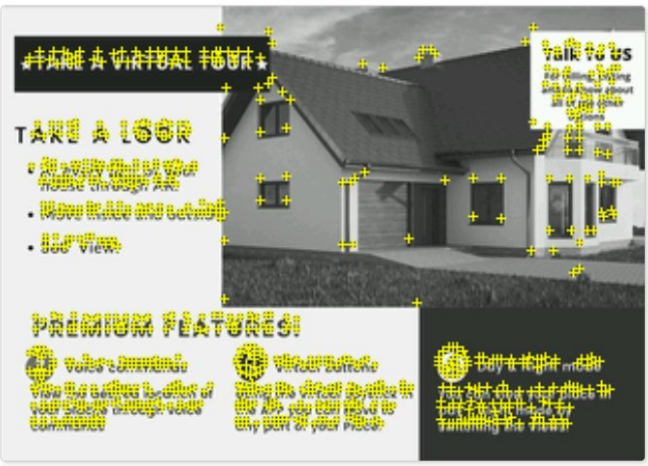


Figure 5.3.1(b3): Target Image Features

4) Download The RIAS Image targets Database:

After checking all these we are good to go. Now download the database by clicking on download button.

5) License Key:

Also copy the License Key from License manager of the data base you want to add to your Unity.

c) Creating 3D house Models:

Here are going to Make 3D house models of the appropriate house images that we have collected and we are going to use Blender 2.6 for that. You can even take it from online if they are available. In the figure below is the example of how a Low-Poly Fully Rigged 3D model looks like.

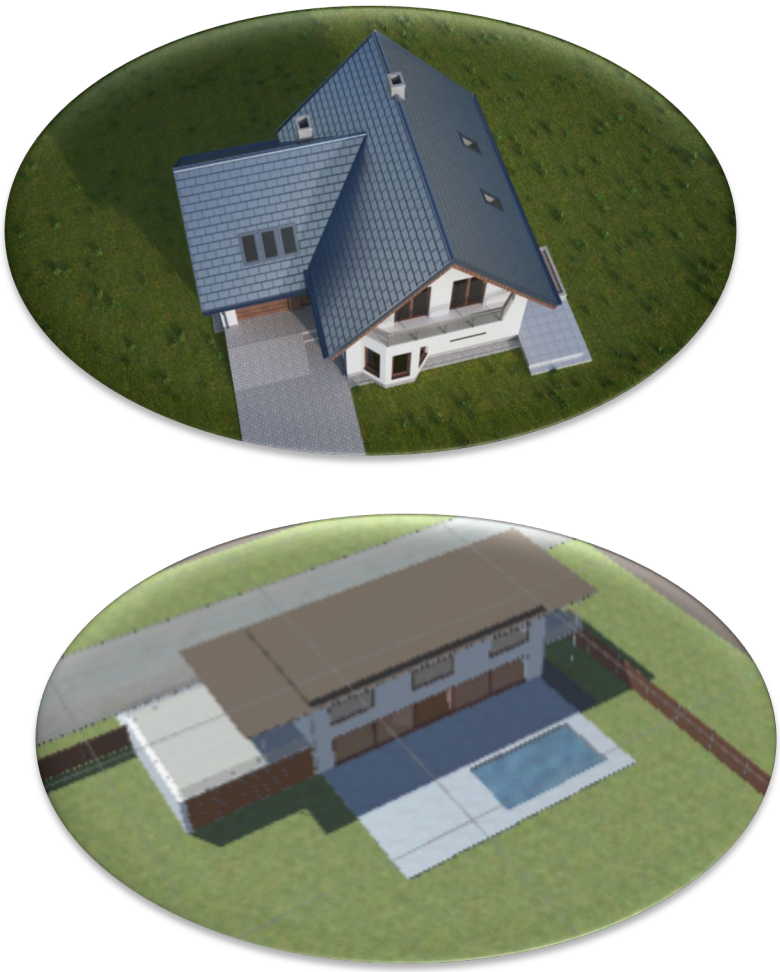


Figure 5.3.1c: 3D Models of House 1 and 2

5.3.1.1 Conclusion:

Now that we pretty much have everything done in the first phase, The first phase completes here!

5.3.2. Integrating with Unity:

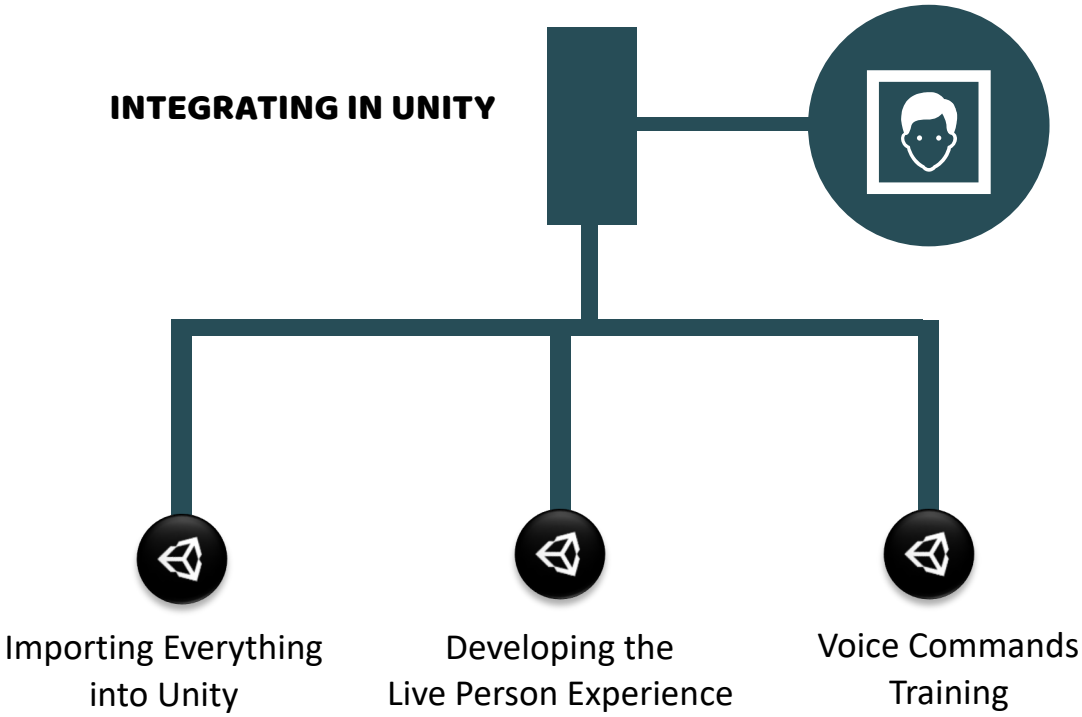


Figure 5.3.2: Methodology Tree Phase 2

a) Importing Everything into unity:

i) Vuforia SDE plugin/Extension: Go to the Build Settings and enable the Vuforia AR tool kit extension so that u have access to the AR camera and all the Augmented Reality Features.

ii) Image targets and its Data Base: The Database of our Target images which we have downloaded in the previous phase along with the Image targets just import them into the unity.

iii) License Key: Take the license key copy it and paste at the engine configuration settings of the AR camera and enable / select the imported database from Inspector panel in Unity.

iv) Import the 3D models: Now import the low-poly fully rigged 3D models into the Unity.

v) AR work: Now Make the 3D house models as prefabs. Later, using Canvas -> UI and 3D models prefabs do the edits/ design the necessary AR graphics that u wants to display in your app.

vi) Coding: To link the game objects, to make the graphics work in Realtime and to support the custom-made featured graphics we need to code them. We are using C# programming to do the coding part. We are going to use Visual Studio to write the C# script and import the After coding attach the appropriate Coding files to the appropriate Game Objects.

vii) FPP Mode: Developing the FPP view using FPP camera and other Game Design Dynamics called Rigid Body, Mesh renderer, Box collider etc., Concentrating on it because it is one of the premium features of the app.

viii) Unity Standard Assets: we are going to download them from unity store and import them into unity. We are going to use Unity dual mobile controls and test it with single mobile control joystick from the unity standard assets pack under unity cross platform assets.

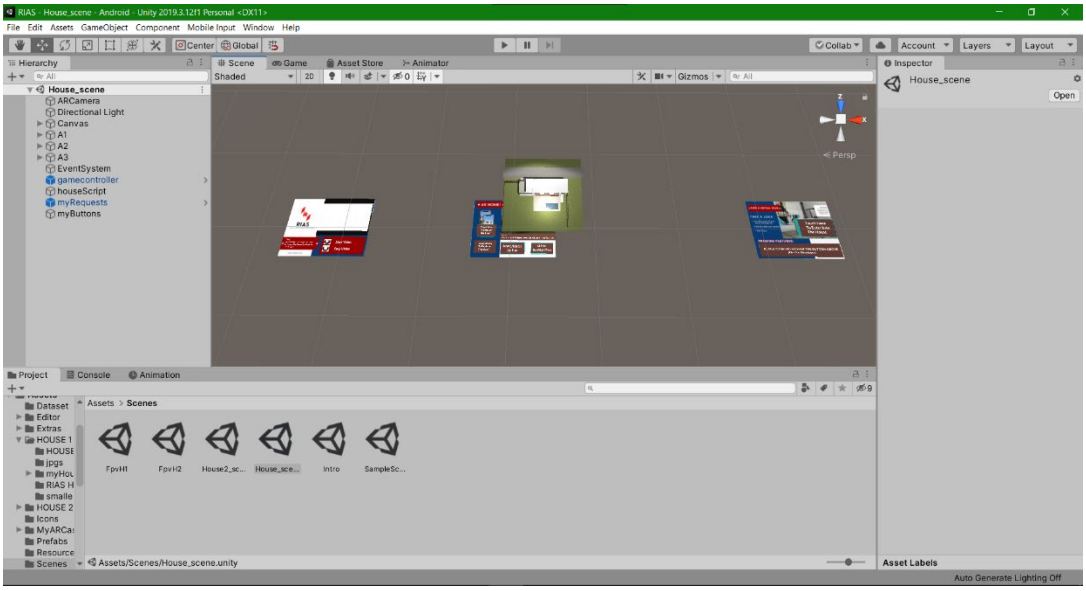


Figure 5.3.2(a): Unity UI - RIAS model development at the end of Phase 2 Step 1

b) Generating the First-person Perspective Experience: This is the Premium feature of our app. We need to first import the 3D House model prefab, import Unity FPP camera from unity standard assets that we have already imported, apply Unity Physics and Dynamics, Add Mobile control Joysticks and give it a test that's all.

i) 3D house Model Prefab: This is the base point or foundation. This 3D model prefab is what responsible to act as a Real 3D house or to generate the AR House 3D graphics when scanned through the camera.

ii) Unity Physics and Dynamics: To be able to move around, inside and outside the house Unity Physics plays a major role. We need to add a few components of Unity Physics Like Rigid body, Mesh renderer, Box collider. And also adding the FPP player camera, Test Joysticks and original Mobile Joysticks.

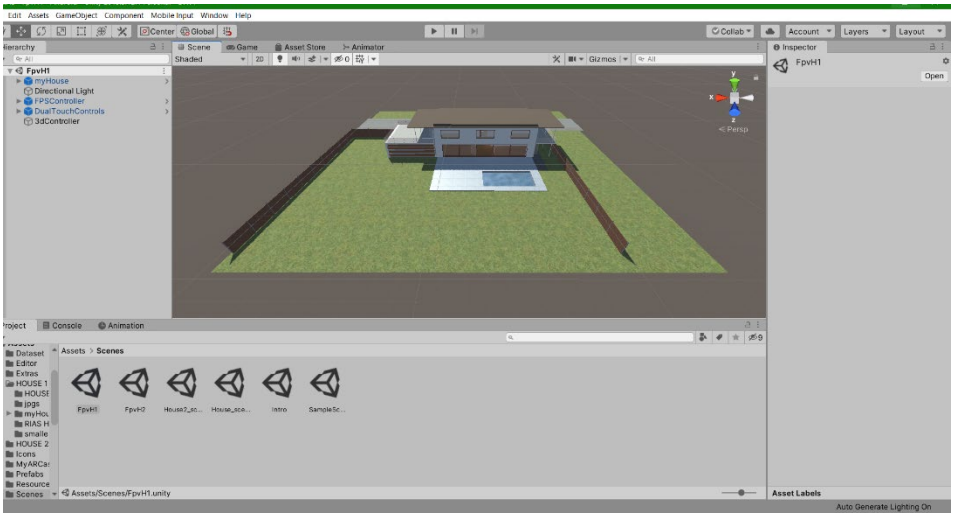


Figure 5.3.2(b1): Unity UI - RIAS model development at the end of Phase 2 Step 2

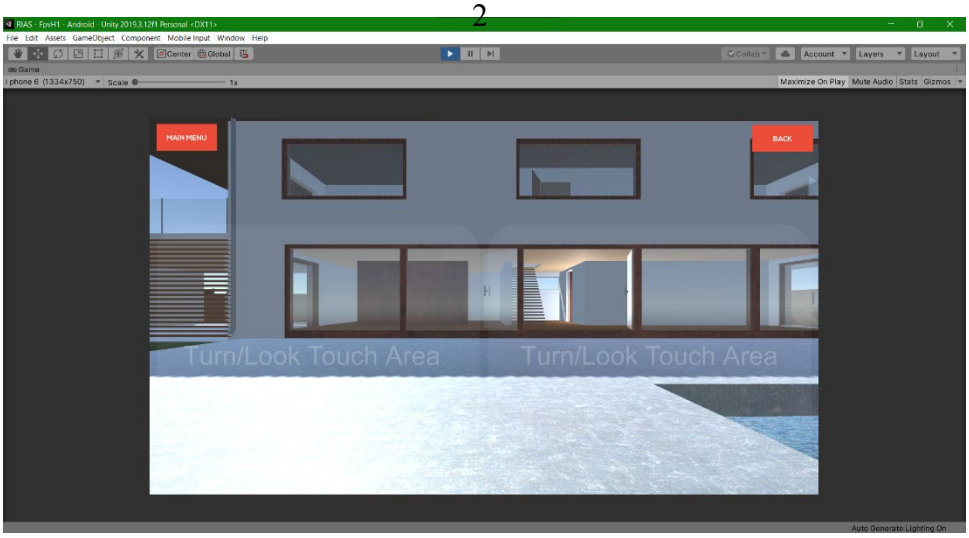
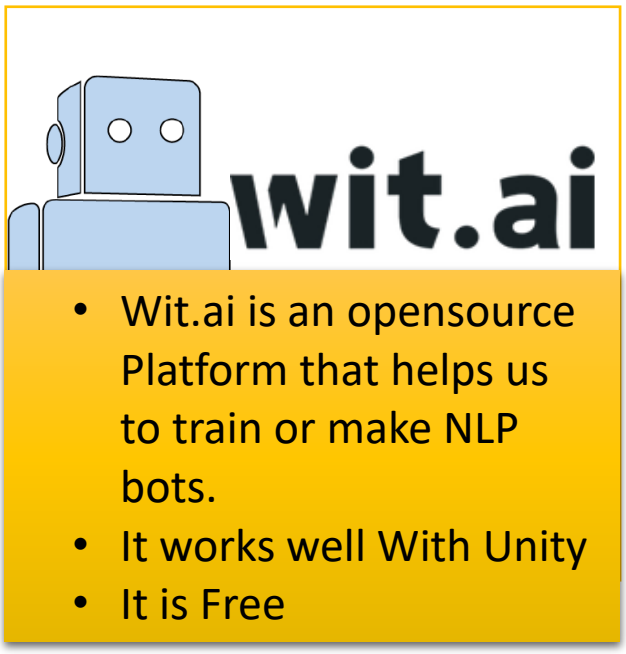


Figure 5.3.2(b2): Unity Game View of Live Home Experience Feature

c) Voice Commands Training:

This is one of the other premium featrures of our app we are going to use the wit.ai platform to develop our NLP bot. after that we are going to integrate it back to unity.



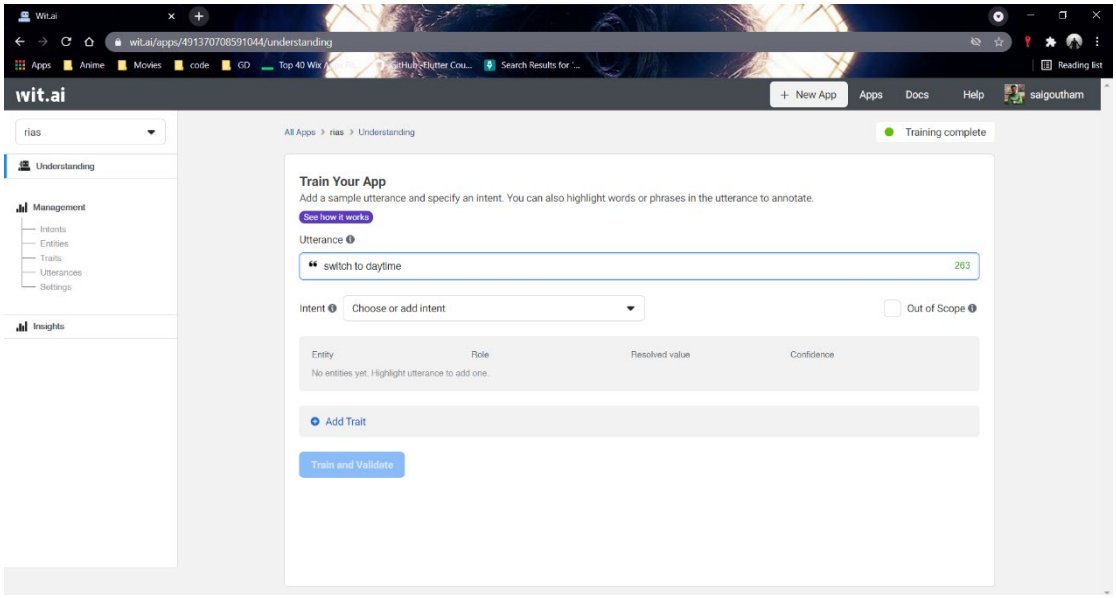


Figure 5.3.2(c): Training RIAS bot from wit.ai

5.3.2.1 Conclusion:

Now that we pretty much have everything done in the second phase, The second phase completes here! And we are building the app to for testing and debugging.

5.3.3. Testing and Debugging:

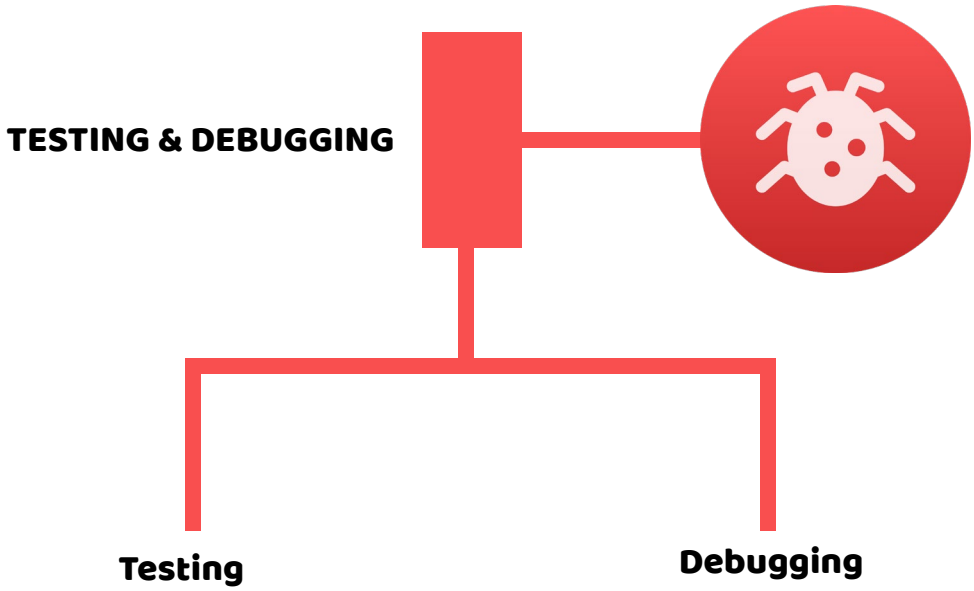


Figure 5.3.3: Methodology Tree Phase 3

We are going to do the following things in this phase:

1. Debugging User Control settings, Target Object Truce Points, Vuforia Engine related common queries.
2. Debug the errors in code.
3. Checking whether the buttons are working properly or not.
4. Fixing the Unity Modelling issues. These usually pop up for (like silhouette model, grey scale, non-rigged models, for perfect 3D models there will be no margin for error.)
5. Improving UI Enhancements and make it more user friendly.
6. Organise the code so that its readable and editable with ease in concern of further Product Development.

5.3.3.1 Conclusion:

We are going to repeat this process until all the bugs are cleared and the UI is neat and clean and all the elements like playing video, buttons works properly, etc., Now that we pretty much have everything done in the third phase, The third phase completes here!

5.3.4. Making an Analysis Report:

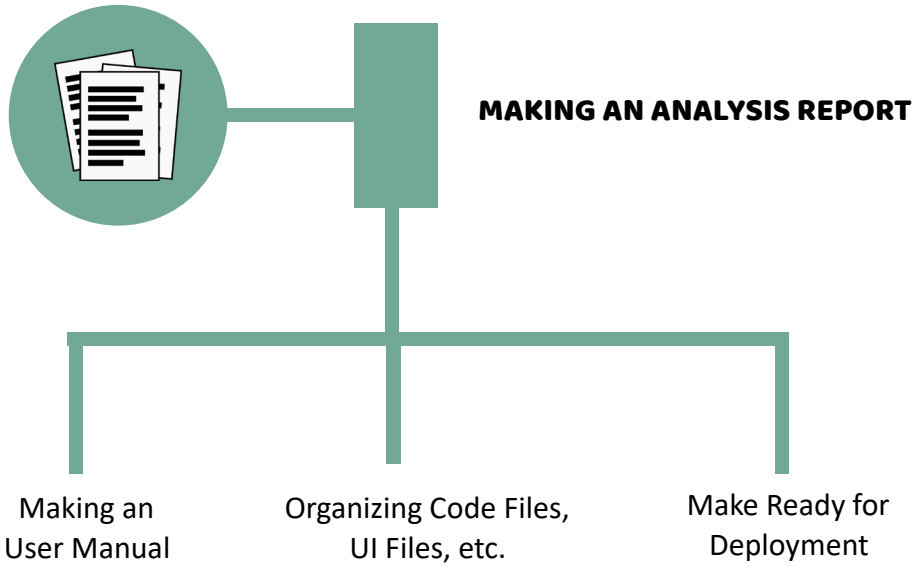


Figure 5.3.4: Methodology Tree Phase 4

5.3.4.1 Conclusion:

This is the Fourth phase and we are now ready to develop the product as an android application and deploy it in the google cloud in the next phase.

5.3.5 Product Deployment:

Figure 5.3.5: RIAS app in Sample Mobile

5.3.5.1 Conclusion:

This is the final phase and we are going to build it as an android application and deploy it in the google cloud.

5.4 RIAS APP INTERFACE:

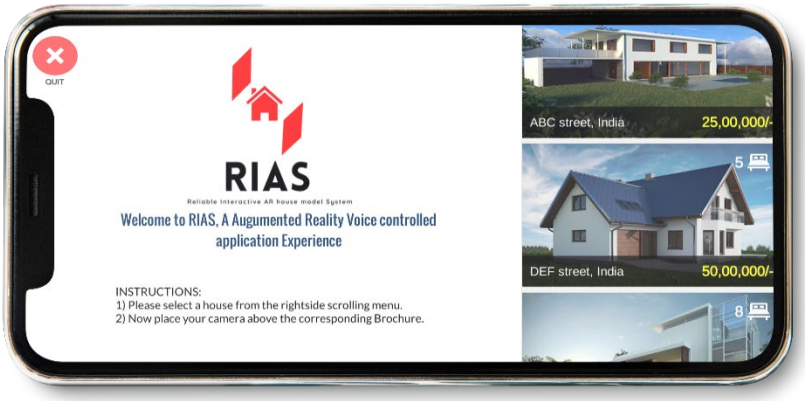


Figure 5.4(a) RIAS app UI – Intro Scene

Figure 5.4(b) RIAS app UI – Image target 1 AR scene

Figure 5.4(c): RIAS app UI – Live Home Experience

5.5 Other Details:

5.5.1 Vuforia target upload process:

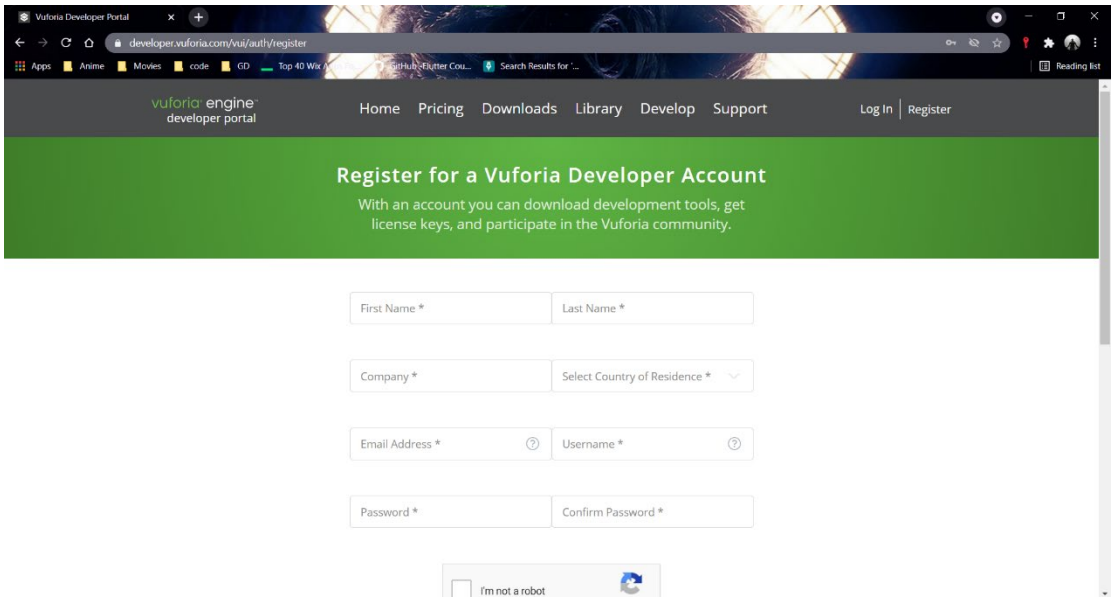


Figure 5.5.1(a): Vuforia Login page

1. Go to Vuforia Developer page and Complete the login info if u don't have an account already then register with the necessary login details.

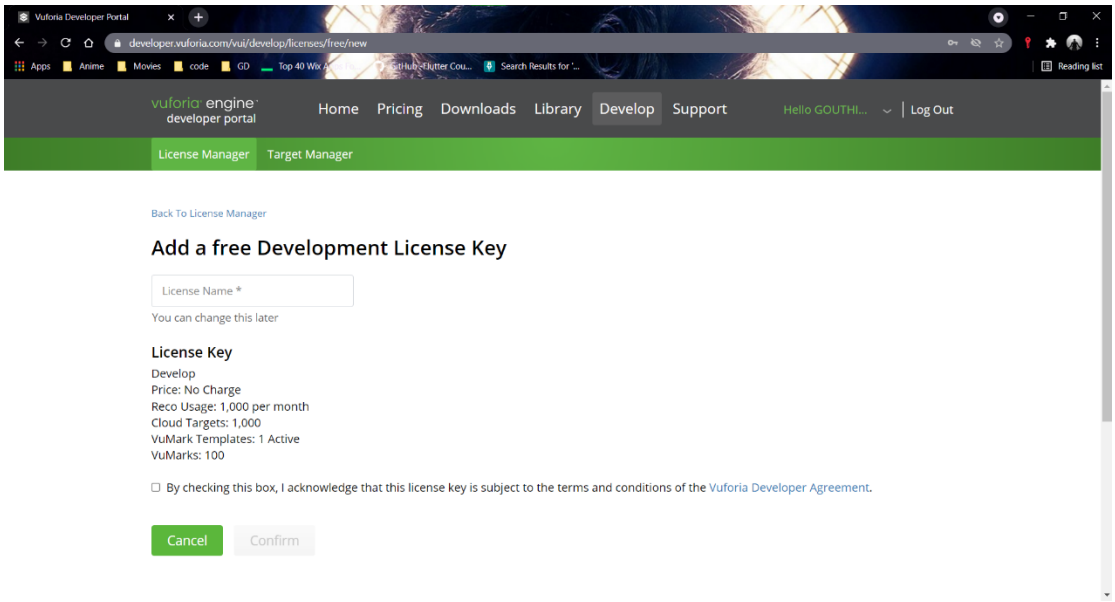


Figure 5.5.1(b): License Key Activation page

2. Create new license key to make a new AR project. This is a very important step because without the license key you cannot get the permissions to use you image target database in Unity.

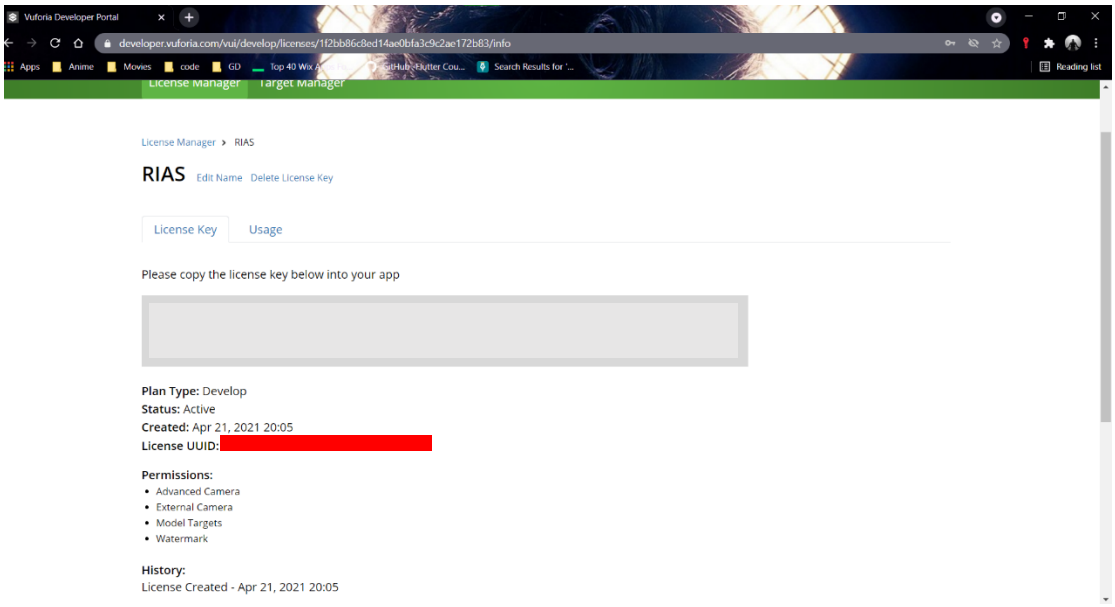


Figure 5.5.1(c): License Key page

3.Copy that license key and enter at the AR camera dialog box from the inspector panel of Unity to get the access rights of the Image target Database.

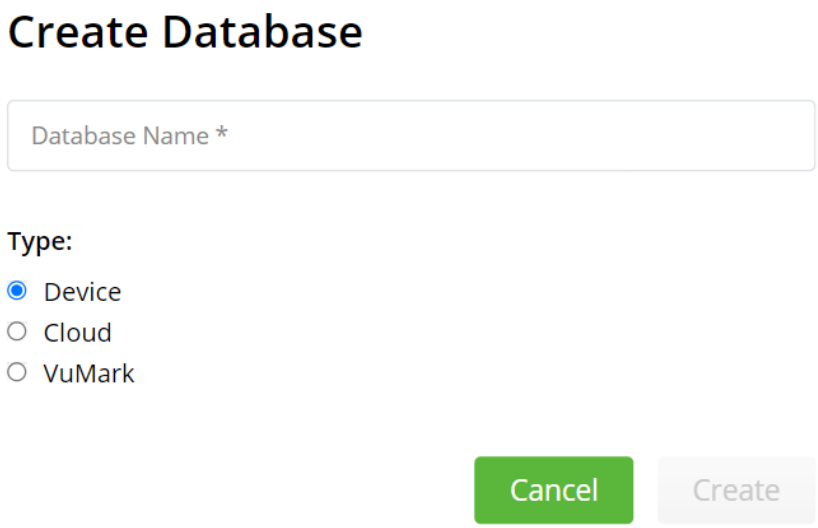


Figure 5.5.1(d): Create Database

4. Now click on Target Manager and create A database using Add Database button. This is where your image targets get saved.

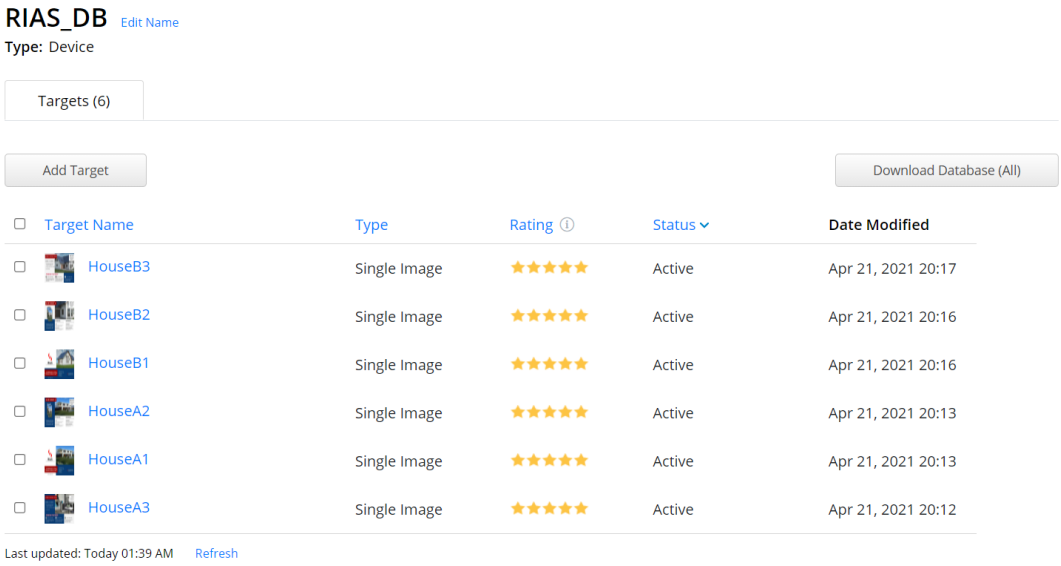
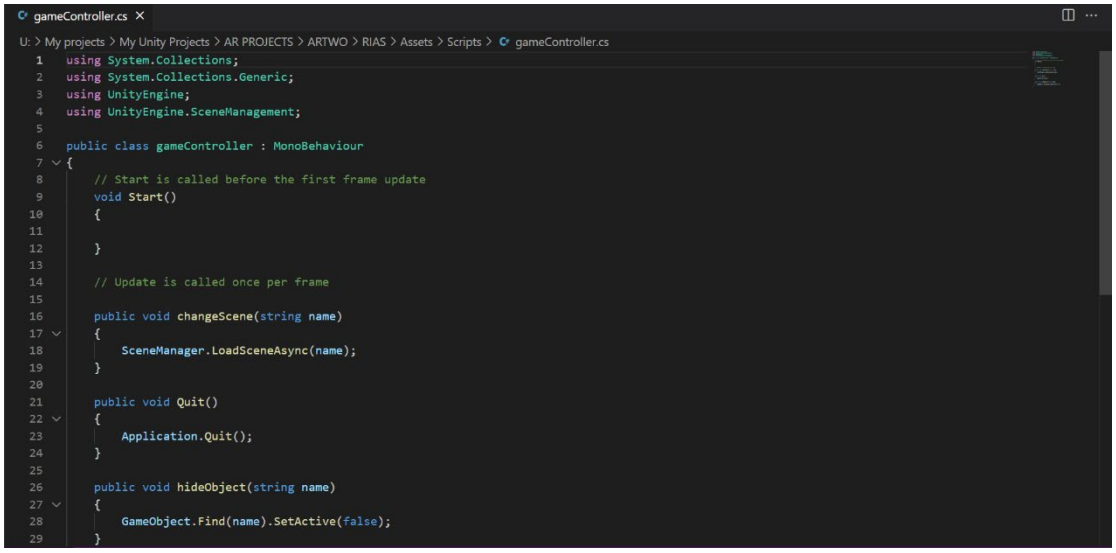


Figure 5.5.1(e): Vuforia Target manager

5. Add the image targets and check the rating. If the ratings are 3 and above96+52, then the image target is said to have good quality. If its below 3 then need to remake and reupload it as bad image targets always have problems while implementation.

5.5.2 Code:



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class gameController : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame

    public void changeScene(string name)
    {
        SceneManager.LoadSceneAsync(name);
    }

    public void Quit()
    {
        Application.Quit();
    }

    public void hideObject(string name)
    {
        GameObject.Find(name).SetActive(false);
    }
```

- ✓ This Script is responsible for functions of buttons – Quit, Main Menu, back.
- ✓ This Script is also responsible for switching the scenes in Unity.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class houseScript : MonoBehaviour
{
    public GameObject theRoof;
    public GameObject[] secondFloor;
    public GameObject[] theLights;
    public GameObject directionalLight;

    public static houseScript instance;
    private bool Daytime = true;

    private void Awake()
    {
        if (instance == null)
        {
            instance = this;
        }
    }
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
```

- ✓ This Script is responsible for the real-time access of house
- ✓ This script is also responsible for the functioning of the Virtual AR buttons

```
    }
    public void togglerooof()
    {
        if (theRoof.activeInHierarchy)
        {
            theRoof.SetActive(false);
        }
        else
        {
            theRoof.SetActive(true);
        }
    }
    public void toggleSecondFloor()
    {
        togglerooof();
        for (int i = 0; i < secondFloor.Length; i++)
        {
            if (secondFloor[i].activeInHierarchy)
            {
                secondFloor[i].SetActive(false);
            }
            else
            {
                secondFloor[i].SetActive(true);
            }
        }
    }
```

- ✓ **togglerooof ( ):**

This function toggles the roof On or off based on the user inputs. In other words, this removes (hides) or replaces (adds) the roof.

✓ **toggleSecondFloor ( ):**

This user-defined function is responsible for removing or replacing the second floor of the house based on the user inputs.

```
public void toggleDayNight()
{
    if (Daytime == true)
    {
        for (int i = 0; i < theLights.Length; i++)
        {
            if (theLights[i].activeInHierarchy)
            {
                theLights[i].SetActive(true);
            }
        }
        directionalLight.GetComponent<Light>().color = new Color32(11, 23, 46, 255);
        Daytime = false;
    }
    else
    {
        for (int i = 0; i < theLights.Length; i++)
        {
            if (theLights[i].activeInHierarchy)
            {
                theLights[i].SetActive(false);
            }
        }
        directionalLight.GetComponent<Light>().color = new Color32(255, 255, 255, 255);
        Daytime = true;
    }
```

✓ **toggleDayNight ( ):**

This user-defined function is responsible for Switching the modes between Daytime or Night time based upon the user inputs.

```
vbButton.cs
U: > My projects > My Unity Projects > AR PROJECTS > ARTWO > RIAS > Assets > Scripts > vbButton.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Video;
using Vuforia;

public class vbButton : MonoBehaviour, IVirtualButtonEventHandler
{
    public GameObject[] myButtons;
    public GameObject confirm_panel;
    public GameObject videoPlane;
    private bool isVideoPlaying = false;
    private VideoPlayer video;
    // Start is called before the first frame update
    void Start()
    {
        for (int i = 0; i < myButtons.Length; i++)
        {
            myButtons[i].GetComponent<VirtualButtonBehaviour>().RegisterEventHandler(this);
        }
        video = videoPlane.GetComponent<VideoPlayer>();

    }

    // Update is called once per frame
    void Update()
    {

```

✓ This script does the entire job of functioning of the Virtual Buttons

The buttons that we have designed are:

1) **Play Button:** To play the video.

2) **Pause Button:** To pause the video.
3) **Day Time / Night Time Switch Button:** To switch to daytime/ Night time.
4) **Replace the Roof Button:** Replaces the Roof if it removed.
5) **Remove the Roof Button:** Removed the Roof if it replaced.
6) **Toggle Second Floor Button:** Toggles the second floor and roof on/off.
7) **Live Home Experience Button:** Take you to FPP view.

vbButton.cs

U: > My projects > My Unity Projects > AR PROJECTS > ARTWO > RIAS > Assets > Scripts > vbButton.cs

```
    }

    public void OnButtonPressed(VirtualButtonBehaviour vb)
    {
        if (vb.name == "buttonT3")
        {
            confirm_panel.SetActive(true);

        }
        if (vb.name == "buttonT1Start" && isVideoPlaying == false)
        {
            video.Play();
            isVideoPlaying = true;
        }
        if (vb.name == "buttonT1Stop" && isVideoPlaying == true)
        {
            video.Stop();
            isVideoPlaying = false;
        }
        if (vb.name == "replaceRoof")
        {
            vb.gameObject.SetActive(false);
            myButtons[3].SetActive(true);
            houseScript.instance.togglerooof();
        }
        if (vb.name == "removeRoof")
        {
            vb.gameObject.SetActive(false);
            myButtons[4].SetActive(true);
            houseScript.instance.togglerooof();
        }
        if (vb.name == "removeSecondfloor")
        {
            vb.gameObject.SetActive(false);
            myButtons[5].SetActive(true);
            houseScript.instance.toggleSecondFloor();
        }
        /*if (vb.name == "replaceSecondfloor")
        {
            vb.gameObject.SetActive(false);
            myButtons[6].SetActive(true);
            houseScript.instance.toggleSecondFloor();
        }*/
        if (vb.name == "dayTime")
        {
            vb.gameObject.SetActive(false);
            myButtons[6].SetActive(true);
            houseScript.instance.toggleDayNight();
        }
    public void OnButtonReleased(VirtualButtonBehaviour vb)
    {

    }

}
```

5.5.3 Unity Engine Concepts:

a) **The Scene View:** The Scene View is where you design the project.
b) **The Game View:** The Game View is rendered from the Camera(s) in your Project. It is representative of your final, published game or project or app. You can simply say that u can test the developed or designed project in the Game view.

c) Mesh Renderer: The Mesh Renderer takes the geometry from the Mesh Filter and renders it at the position defined by the Game Object's Transform component.

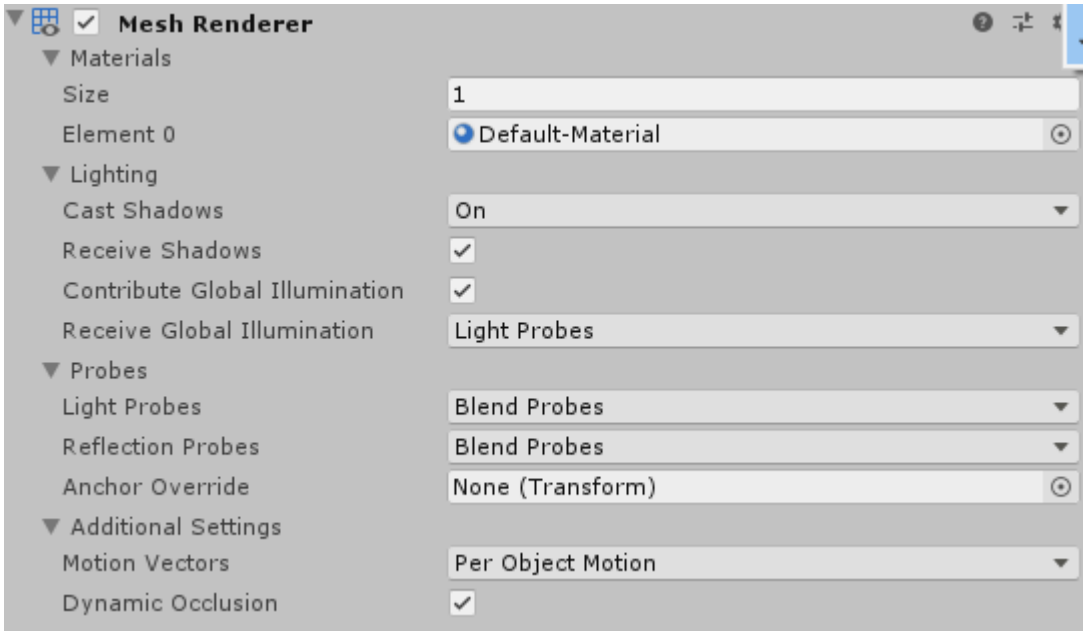


Figure 5.5.3: Mesh Renderer

d) Materials: The Materials section in the Mesh Renderer Inspector lists all the Materials that the Mesh Renderer is using. Meshes imported from 3D modelling software can use multiple Materials, and each sub-Mesh uses one Material from the list.

If a Mesh contains more Materials than sub-Meshes, Unity renders the last sub-Mesh with each of the remaining Materials, one on top of the next. This allows you to set up multi-pass rendering on that sub-Mesh. However, this can impact the performance at run time. Fully opaque Materials overwrite the previous layers, which causes a decrease in performance with no advantage.

e) Lighting: The Lighting section contains properties for how this Mesh Renderer interacts with lighting in Unity.

f) Light Mapping: This section is visible only if only if Receive Global Illumination is set to Lightmaps.

g) Rigid body: Control of an object's position through physics simulation.

Adding a Rigid body component to an object will put its motion under the control of Unity's physics engine. Even without adding any code, a Rigid body object will be pulled downward by gravity and will react to collisions with incoming objects if the right Collider component is also present.

The Rigid body also has a scripting API that lets you apply forces to the object and control it in a physically realistic way. For example, a car's behavior can be specified in terms of the forces applied by the wheels. Given this information, the physics engine can handle most other aspects of the car's motion, so it will accelerate realistically and respond correctly to collisions.

In a script, the Fixed Update function is recommended as the place to apply forces and change Rigid body settings (as opposed to Update, which is used for most other frame update tasks). The reason for this is that physics updates are carried out in measured time steps that don't coincide with the frame update. Fixed Update is called immediately before each physics update and so any changes made there will be processed directly.

A common problem when starting out with Rigid bodies is that the game physics appears to run in "slow motion". This is actually due to the scale used for your models. The default gravity settings assume that one world unit corresponds to one meter of distance. With non-physical games, it doesn't make much difference if your models are all 100 units long but when using physics, they will be treated as very large objects.

If a large scale is used for objects that are supposed to be small, they will appear to fall very slowly - the physics engine thinks they are very large objects falling over very large distances. With this in mind, be sure to keep your objects more or less at their scale in real life (so a car should be about 4 units = 4 meters, for example).

h) Canvas: The Canvas is the area that all UI elements should be inside. The Canvas is a Game Object with a Canvas component on it, and all UI elements must be children of such a Canvas.

Creating a new UI element, such as an Image using the menu Game Object > UI > Image, automatically creates a Canvas, if there isn't already a Canvas in the scene. The UI element is created as a child to this Canvas.

The Canvas area is shown as a rectangle in the Scene View. This makes it easy to position UI elements without needing to have the Game View visible at all times.

Canvas uses the Event System object to help the Messaging System.

- **Draw order of elements:**

UI elements in the Canvas are drawn in the same order they appear in the Hierarchy. The first child is drawn first, the second child next, and so on. If two UI elements overlap, the later one will appear on top of the earlier one.

To change which element appear on top of other elements, simply reorder the elements in the Hierarchy by dragging them. The order can also be controlled from scripting by using these methods on the Transform component: SetAsFirstSibling, SetAsLastSibling, and SetSiblingIndex.

- **Render Modes:**

The Canvas has a Render Mode setting which can be used to make it render in screen space or world space.

a) Screen Space - Overlay

This render mode places UI elements on the screen rendered on top of the scene. If the screen is resized or changes resolution, the Canvas will automatically change size to match this.

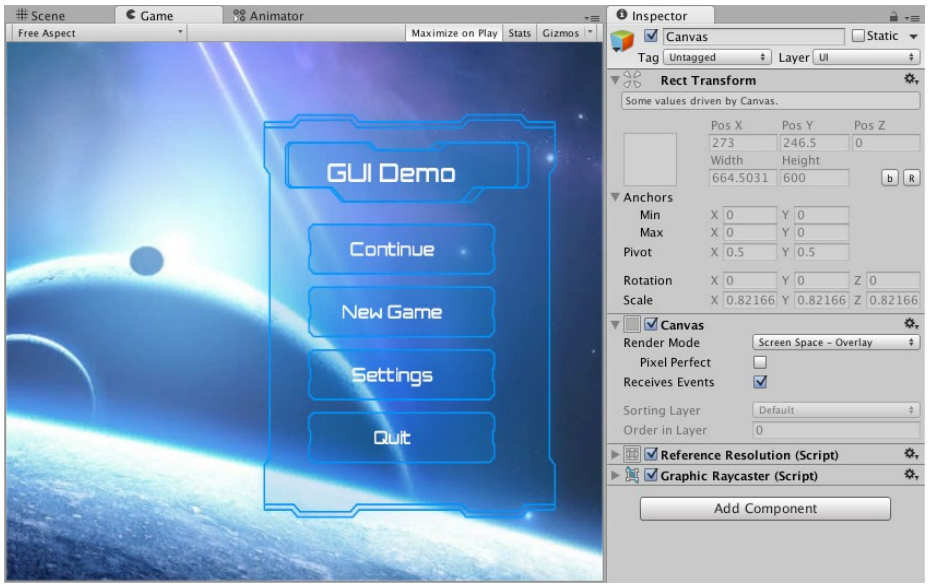


Figure 5.5.3(h-a): Screen Space - Overlay

b) Screen Space - Camera

This is similar to Screen Space - Overlay, but in this render mode the Canvas is placed a given distance in front of a specified Camera. The UI elements are rendered by this camera, which means that the Camera settings affect the appearance of the UI. If the Camera is set to Perspective, the UI elements will be rendered with perspective, and the amount of perspective distortion can be controlled by the Camera Field of View. If the screen is resized, changes resolution, or the camera frustum changes, the Canvas will automatically change size to match as well.

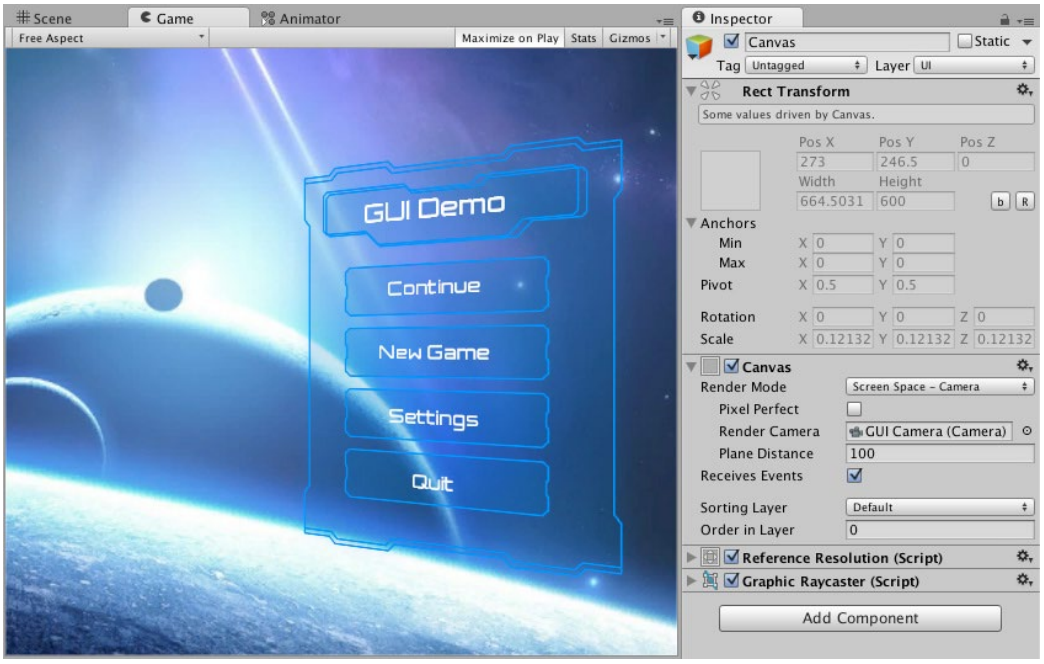


Figure 5.5.3(h - b): Screen Space - Camera

c) World Space

In this render mode, the Canvas will behave as any other object in the scene. The size of the Canvas can be set manually using its Rect Transform, and UI elements will render in front of or behind other objects in the scene based on 3D placement. This is useful for UIs that are meant to be a part of the world. This is also known as a "diegetic interface".

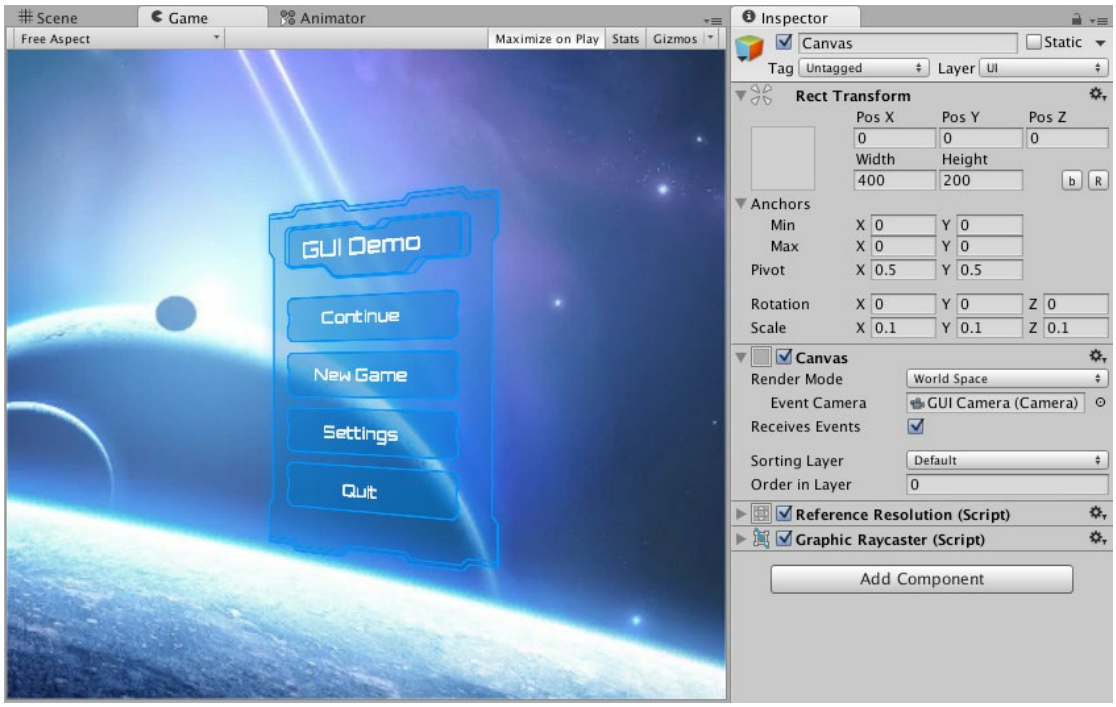


Figure 5.5.3(h - c): World Space - Camera

5.5.4 Vuforia AR SDK:

Vuforia is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to recognize and track planar images and 3D objects in real time. This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real world objects when they are viewed through the camera of a mobile device. The virtual object then tracks the position and orientation of the image in real-time so that the viewer's perspective on the object corresponds with the perspective on the target. It thus appears that the virtual object is a part of the real-world scene.

The Vuforia SDK supports a variety of 2D and 3D target types including ‘markerless’ Image Targets, 3D Model Target, and a form of addressable Fiducial Marker, known as a VuMark. Additional features of the SDK include 6 degrees of freedom device localization in space, localized Occlusion Detection using ‘Virtual Buttons’, runtime image target selection, and the ability to create and reconfigure target sets programmatically at runtime.

Vuforia provides Application Programming Interfaces (API) in C++, Java, Objective-C++, and the .NET languages through an extension to the Unity game engine. In this way, the SDK supports both native development for iOS,

Android, and UWP while it also enables the development of AR applications in Unity that are easily portable to both platforms.

Vuforia has been acquired by PTC Inc. in November 2015.

AR CAMERA:

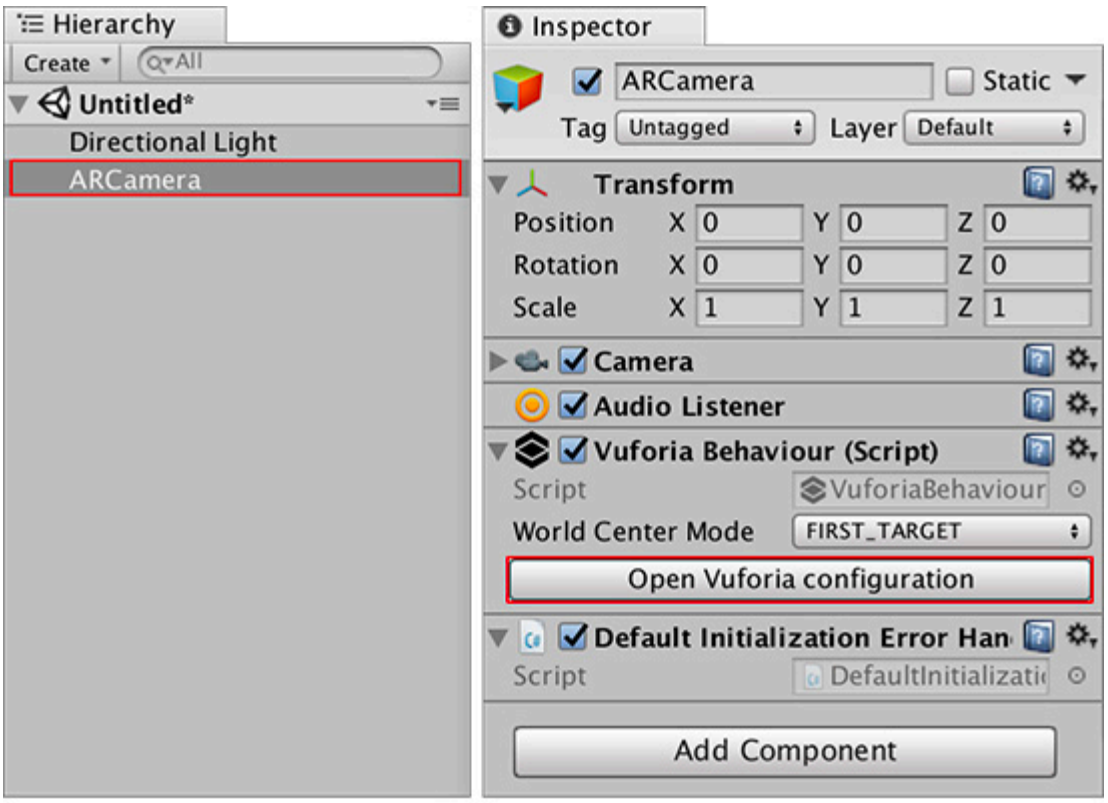


Figure 5.5.4: AR Camera Configuration

Current Market AR SDK's:

We can define several significant differences in the basic performance of most AR-based apps. Thus, some solutions focus on identifying graphic markers (like QR-codes or complex graphs), others process data received via GPS while the rest may combine both features. An emerging technology is a full-body motion tracking AR/VR solutions for those in the industries who require believable real-world interactions inside their virtual world environments.

So, among all of the 10 best available AR SDK's jelvix.com ranked that Vuforia's AR SDK as no.1. The comparisons are shown below.

These are the best available AR SDKs suggested by Jelvix.com

	vuforia	wikitude	ARKit 3	ARCore	MRXT
Platforms	Android, iOS, UPW and Unity Editor	Android, iOS, Windows for tablets	iOS 11+	Android 7.0+, iOS 11+	Android, iOS, Windows and Mac OS
License Type	Free, Commercial	Commercial	Free	Free	Free, Commercial
Smart Glasses Support	+	+	+	+	+
Unity Support	+	+	+	+	+
Cloud Storage	+	+	+	+	-
3D Recognition	+	+	+	+	+
Geolocation	+	+	+	+	-

	EasyAR SDK 2.0	kudan	ARTOOLKIT	Armedia	DEEPAR.AI
Platforms	Android, iOS, UPW, Unity Editor, Windows and Mac OS	Android and iOS	iOS, Android, Windows Linux, SGI and MAC OS X	iOS, Android, Windows and Flash	iOS, Android, Windows and WebGL
License Type	Free	Free, Commercial	Free	Free, Commercial	Unknow
Smart Glasses Support	+	+	+	+	-
Unity Support	+	+	+	+	+
Cloud Storage	+	+	-	+	-
3D Recognition	+	+	+	+	+
Geolocation	+	+	+	+	-

Jelvix jelvix.com

Table 5.5.4: AR SDKs Comparison table

5.5.4 Blender 2.6

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, virtual reality, and computer games.

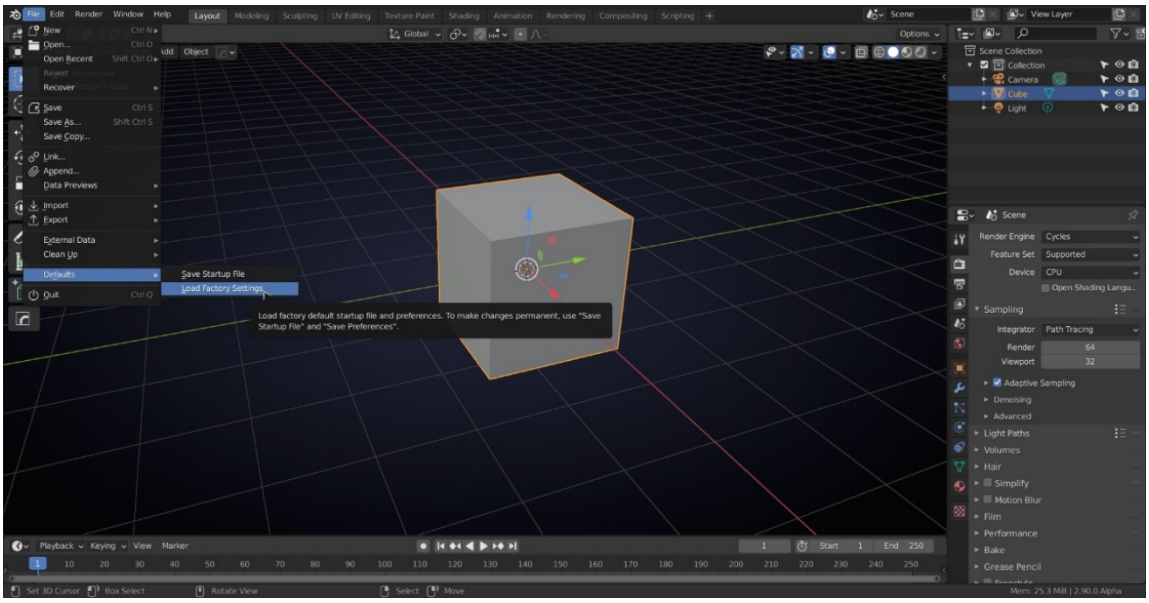


Figure 5.5.4(a): Blender 2.6 Interface

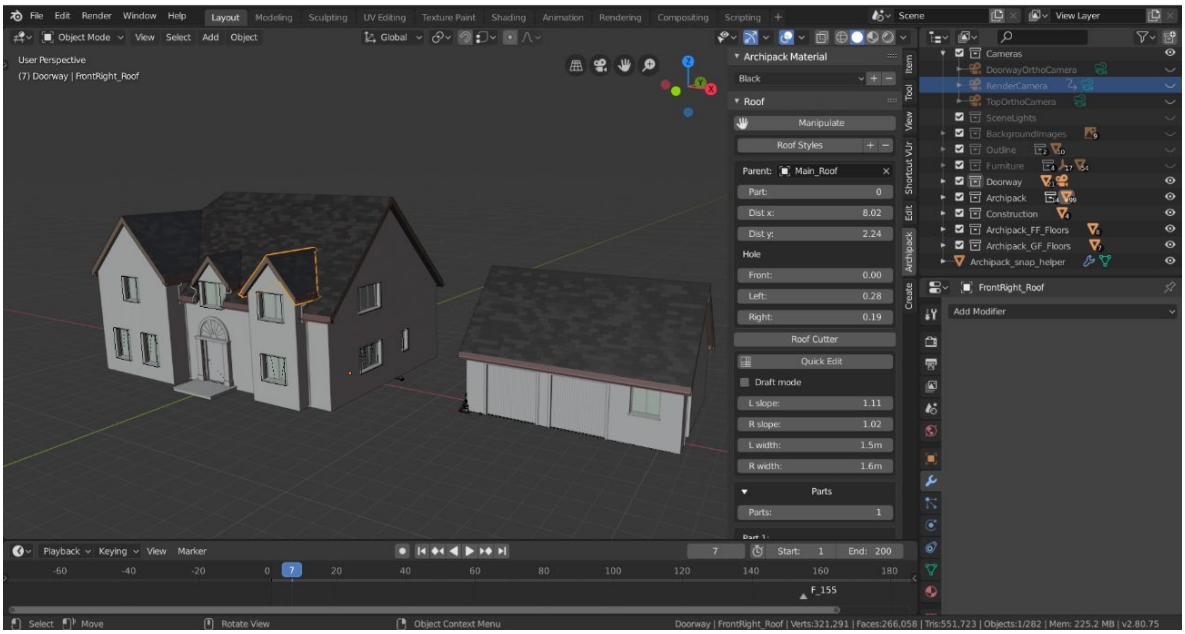


Figure 5.5.4(b): House model-2

5.5.5 Adobe Photoshop 2021:

Adobe Photoshop is a raster graphics editor developed and published by Adobe Inc. for Windows and macOS. It was originally created in 1988 by Thomas and John Knoll. Since then, the software has become the industry standard not only in raster graphics editing, but in digital art as a whole.

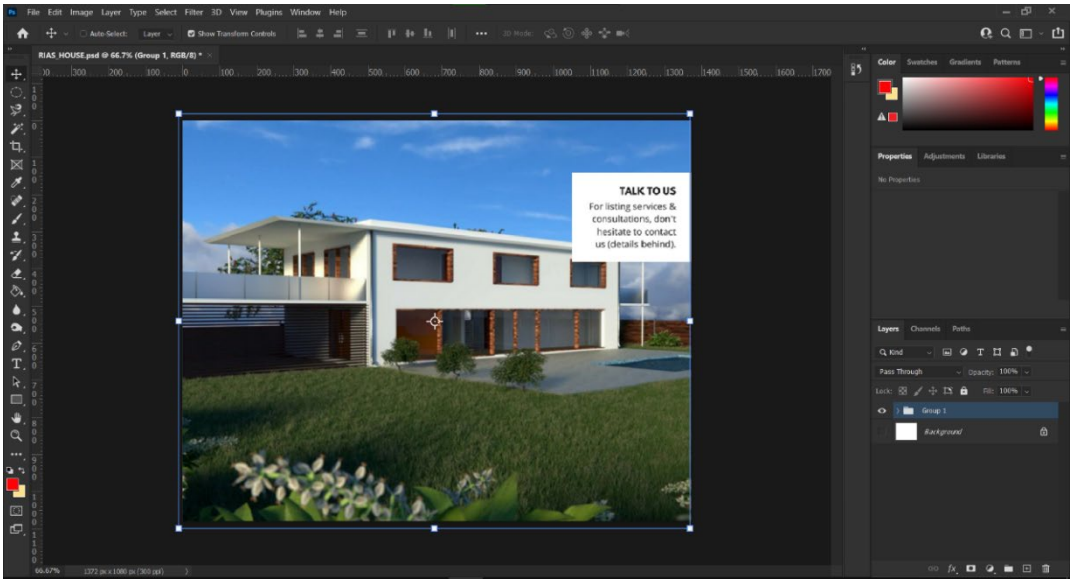


Figure 5.5.5: Adobe Photoshop 2021 Image Target Design

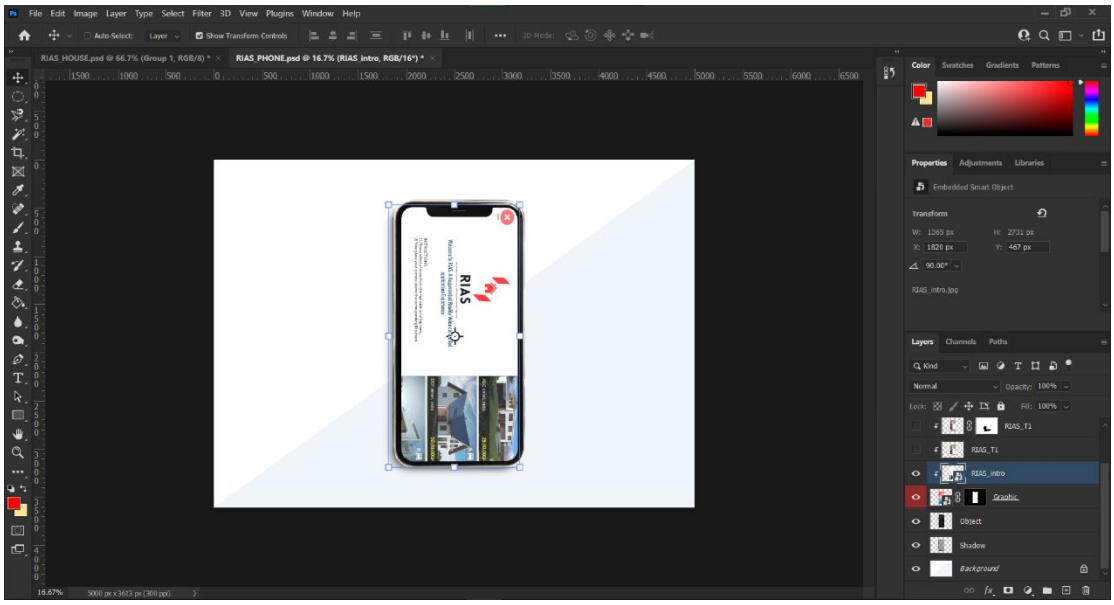


Figure 5.5.5(b): Adobe Photoshop 2021 App UI

6. CONCLUSION

6.1 Project Contribution:

RIAS is designed to Public, which will completely revolutionize the Digital Experience and marketing approach in the field of Real Estate through the latest technology Augmented Reality, RIAS user-friendly Interface allows users to use it more as it is more interesting, attractive and provides a real-time experience.It's Because Our Project has been successful in making the user's ideology express through AR representation to the customers instead of using the paper card and able to bring new Real-world Experience through our app's premium features Live Home Experience. It helps both to the seller and the customer to have a whole new Interactive, Attractive and Safest ever Experience by going through the house they desired to buy right from their own mobile screens.

6.2 Product Usage:

To Experience RIAS in AR, users need to just download the RIAS app from the google cloud link that has been provided on the brochure and the simply scan a card to enjoy the animation, sounds and interact with buttons.

6.3 Future Enhancements:

- Although the application operates in properly and follows the milestone of project respectively, there exist some changes can be taken for better performances.
- Based on the Requirements By downloading the data from the clouds Interested people can add further Enhancements to the App. Interested people can also integrate Other AI bots for more features.

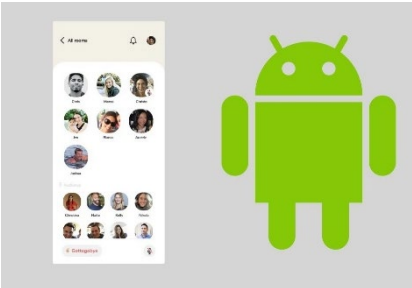


Figure 6.3 Future Enhancements

- One of the good future enhancements can be Inserting chat rooms to directly chat with the dealer and existing residents

6.4 Applications:

1. REAL ESTATE *like RIAS (our App).*
2. HEAVYLY CROWDED PUBLIC PLACES *like Malls, etc,*
3. PLACES WHERE MAP IS REQUIRED *like Zoo Parks, Big Parks, Adventure Parks, Comic-cons etc,*
4. UNIVERSITIES *like to show the routes to cafeteria, Principal Office, etc., for newcomers.*
5. FOR MULTISTORYED BUILDINGS & SKYSCAPERS *like in-case of fire accidents we can trace people.*
6. FOR CIVIL ENGINEERS & ARCHITECTURE *like to understand Floor-plans in 3D, etc,*

6.5 RIAS vs Existing system Models:

RIAS	Existing System Models
Playing videos, Animations, Interactions, etc. are possible.	It's not possible unless they replicate something like RIAS.
Premium Features like Voice commands Control, Live Home Experience are available as RIAS is the first app to develop those features.	No Special Features Available. Has a boring natural paper view.
It is Free and Easy to use.	Depends on the developer.
RIAS is designed to Public.	Depends on the developer.

Table 6.5: RIAS vs Existing System Models

6.6 Conclusion:

In conclusion, at every new reflection in communication technology, the marketing and advertising community brought their existing models with them in their first efforts to adapt the new medium. Radio, Television, Internet, Web and Mobile have all seen this progression playout AR will likely follow that model too.

The new AR ecosystem might not happen in all the ways for seen but it definitely will happen in some of those ways. This next big thing in the marketing is sure to render obsolete many existing marketing and advertising business models and replace them with new ones.

7. BIBILOGRAPHY

https://www.geeksforgeeks.com

https://www.wit.ai.com

https://www.wikipedia.com

https://www.linkedinlearning.com

https://www.youtube.com/user/Brackeys

https://unity.com

https://developer.vuforia.com

https://www.arvr.google.com

https://www.developer.google.com

https://www.augment.com

https://www.nptel.co

https://geeksforgeeks.com

https://stackoverflow.com

https://udemy.com

https://coursera.com

https://google.com

www.medium.com

References:

1. AR Gamer: Discover Voice Controlled AR Apps - A course by Diego Herrera
2. Augmented Reality: Innovative Perspectives across Art, Industry, and Academia by John Tinnell and Sean Morey.
3. The Fourth Transformation by Robert Scoble and Shel Israel.
4. Complete Virtual Reality and Augmented Reality Development with Unity: Leverage the power of Unity and become a pro at creating mixed reality applications by Jesse Glover and Jonathan Linowes.
5. Unity 2018 Augmented Reality Projects: Build four immersive and fun AR applications using AR Kit, AR Core, and Vuforia by Jesse Glover.